Computer History Museum

Oral History of Dennis R. Austin

Interviewed by:
David C. Brock

Recorded: March 31, 2015
Mountain View, California

CHM Reference number: X7445.2015

David C. Brock: Well, Dennis, I thought we could start at the beginning, as it were, and talk about when and where you were born, and a little bit about your family.

Dennis R. Austin: Well, although I've been a long-time Californian now I'm actually from Pittsburgh, Pennsylvania. I have no family there anymore, although I do still go to a community in far Western New York, Chautauqua, every summer for a week, which is a kind of vacation community in the arts and so forth, but that's my only real attachment to the East anymore. I moved here in '73.

Brock: Did you grow up through high school in Pittsburgh?

Austin: Yes. I went to The University of Virginia for my undergraduate education, and came to California soon after graduation. I managed to get married first to a woman I met in college, and I continued my education in graduate school, but I went to several different schools, and I never managed to finish any of them. I went to Arizona State University. I was in Phoenix for a while. I went to MIT when I was in Boston for a while, and then I was in Santa Barbara. I went to UC Santa Barbara for a while, but all of these I left before I could finish a degree, so eventually I gave up with about 50 credit hours of graduate school. So that's the way I ended up.

Brock: Were you going to these particular places to work with particular people? Is that what was…

Austin: No, because I was working at companies in these places. I was with General Electric which used to be in the computer business in Phoenix, Arizona, and that's when I went to ASU. That was acquired by Honeywell. They bought GE's computer business, and Honeywell promptly moved me to Boston. This was after only a year or so. Then I was in Boston for two years, and I went to MIT and got married. Then I moved to Santa Barbara where I really started my career longer term with Burroughs Corporation, and that was UC Santa Barbara.

Brock: To return to Pittsburgh just for a little bit, was there anything in your parents' activities that you connect to your later involvement with technical matters, or was it a technical household? Were you following a family tradition, or doing something new?

Austin: Definitely not. My father was kind of a musician, performer and salesman sort of guy, and had many different jobs in his career. That seemed to have skipped a generation. Neither my brother nor I did either of those things, but my son is an actor, so it came back out. He's making at least some money at it, probably more than my father did.

Brock: Did you attend the public schools in Pittsburgh?

Austin: Yes. I was in a small community called Rosslyn Farms, which is about halfway between downtown and the airport on the west side of town. Rosslyn Farms had a school of its own, and we had eight grades, four classrooms, two classes per grade. There were 11 children in my class, which was about average. Some were as high as 13 for the bigger classes, but it was a very small school, and I think it was an excellent education. We could walk to school. One of the four teachers served as principal, plus a janitor, and that was the whole staff.

Brock: How did you find school? Did your studies come easy to you?

Austin: Yes. I really liked school pretty much, especially when I got to seventh and eighth grade when we were starting to get to things like algebra. It was such a small school that it was easy to study independently. The teacher could give me assignments in advance in the book, and I could just start getting ahead, and that was a lot of fun.

Brock: Was it math that really attracted you at that time?

Austin: Yes it was, yeah, in particular. We didn't have a science teacher because we only had the general teachers, but we did have science lab equipment and stuff like that that I could experiment with.

Brock: As you got into high school, how did your interests evolve during those times?

Austin: I became convinced that I would probably be an engineer. I was a member of the Future Engineers of America and so forth. I got very good grades. I was very social. I was in a lot of clubs, and I was in the band. When you went to high school from Rosslyn Farms you'd go to the neighboring town which was Carnegie, Pennsylvania which was pronounced car-NEGG-ee by Andrew Carnegie, not CARnegie as they say in New York. In Carnegie High School the band emulated the band of what was then Carnegie Tech and now Carnegie Mellon University, which wore kilts in the band, and they had bagpipers out front, and I was a bagpiper in high school.

Brock: Did you pick that up in high school?

Austin: I did because a lot of the kids that had been through the school system in Carnegie had learned band instruments in school at earlier years, but we didn't have that at little Rosslyn Farm School. I'd taken some piano lessons, but that didn't help with band. I picked up a little guitar, and that didn't help with band either, but they said, "Do you want to learn the bagpipes?" So I wasn't behind in bagpipes. I was right there at the front, so I learned bagpipes. Unfortunately, I have not played since, but I always like to hear bagpipes.

Brock: You can't really be a retiring person when you're blowing bagpipes.

Austin: Right.

Brock: The idea that you would be an engineer, just the idea for somebody of that age, I would imagine, would be either engineers in your community, which I'm sure there were plenty of in the Pittsburgh region, or through some sort of an experience of something that you read, or in the popular media, in the popular culture. Do you know where it came from for you?

Austin: It was probably sort of popular culture, but it may have been largely my own imagination. The teachers in high school talked to you about what different career paths are, but I really liked the idea of creating things. That's what fascinated me about engineering. I always admired arts, and I liked the creative aspect of arts, but I never could seem to find any talent in it to do that. My girlfriend in high school was an artist, and actually had an artistic career after that. I always loved the arts and I still do, but I can't do it. I could do engineering, I thought.

Brock: How did you decide on The University of Virginia?

Austin: I was looking at mostly technical schools, even Carnegie Tech, which was in Pittsburgh. Incidentally, Carnegie Tech at that time, since it wasn't a university yet as it is now, really had two major tracks. One was engineering and sciences and the other was fine arts. They had a great theater program, for example, and art history and things like that. Again, I liked that combination, so that was an attractive school, but I really didn't want to stay in Pittsburgh. I was accepted there, but I just decided against it. The University of Virginia came up as kind of a surprise thing. I just read about it was a very different sort of school. It was a men's school at the time only, which was unusual for a state school. That was actually a negative for me, but I could live with it. It had lots of traditions which were interesting, and I kind of liked oddball things like all the students wore coats and ties all the time. That was the only school in the country like that, I'm sure. It had a great honor tradition, and a lot of other interesting things that attracted me to it, and I was a big admirer of Thomas Jefferson. He was always a hero of mine, and he founded The University of Virginia. It's in the same town as his home was in Charlottesville, Virginia. It was just the right amount of distance away that I was sort of cutoff from my own back in Pittsburgh, and yet close enough that you could drive in a day easily. So I liked the distance too. Shockingly, at the time, it was a completely different culture for being such a short distance away. I went from a steel mill town to the agricultural south.

Brock: What year did you enter?

Austin: Sixty-five.

Brock: It's just an interesting choice given the times at large in the country. It's just kind of an interesting choice for something to choose tradition, to select a place that was all men and everybody wearing a coat and tie. Have you ever reflected on that given what was starting to boil up in the culture at large that you were attracted to this different kind of…

Austin: I think it was just the unusualness of it. I've always looked for unusual. I don't know, that's been me for a long time. So I tried it out, and it was very appealing to me. I made some good friends there that are still friends of mine. I'm going on a trip to Europe in May with a guy I met the first year at Virginia, and who I see fairly regularly, but I have not lived in the same town with for almost 50 years.

Brock: What did you find in terms of an engineering education when you got there?

Austin: Well, I found that I liked it. I found it was much harder than I had thought it was going to be. High school had been pretty easy, and I got good grades, and I thought, well, I pretty much had the same attitudes for what it would be like in college, and it was not that easy. I could have skipped the first semester of calculus, because I had had that in high school. I elected to go ahead and take the regular course, and I was so glad I did because after four weeks they'd covered everything I knew in calculus, and I was already falling behind. So I really had to start studying harder, but it was a good school. I liked the professors. I liked everything about it.

Brock: Was it during those years in Virginia that you first encountered computers, or had you encountered them before?

Austin: My computer experience started in high school with a program by the National Science Foundation for high school kids that, again, was at Carnegie Tech, which at the time had a Bendix G15 computer, and they eventually got two of them, I think. The museum has a G15 in its collection. I look at it fondly. The unusual thing about it is that it had an ALGOL compiler, and ALGOL was a brand new language just a couple of years old. It was probably the first language that was invented purely and that nobody had quite figured out how to write a compiler for it when that came into existence. 1960 was when ALGOL came out. They had done an ALGOL compiler for the Bendix G15, so that was my first language, which is much unlike every other student that I know of. Everybody started with FORTRAN, or assembly language, or something like that. When I went to the University of Virginia they had a Burroughs B5000, B5500 is the model they had, which was a very unusual computer at the time in that it did not have an assembly language, and all the systems software was written in ALGOL.

Brock: Just by happenstance.

Austin: Yeah, a coincidence for me, so that when I started programming in college it was, again, in ALGOL. Burroughs had their own version of ALGOL with a few modifications, but they had a very unusual machine design such that when it came to study how computers actually worked they kind of deemphasized it in the class how the B5500 worked. Instead we used an emulator that emulated an IBM 7090 which was a more standard sort of machine, and we could write code for the 7090 and run it in emulation. But I was kind of fascinated by ALGOL and high-level languages in general, and I ended up sort of staying in high-level languages for a lot of my early career.

Brock: The NSF program in high school to give you that experience -- that was a program to expose high school students to computers and computer languages and programming?

Austin: There were two phases of it. Part of it was electrical. We actually did stuff in the electrical engineering lab at Carnegie Tech. We built things and learned a little about circuits. I knew something about circuits. I was a ham radio operator in high school too, so electronics certainly wasn't new to me, but we had that section and then we also had the computer section. As I recall it was mornings in the computer lab, and afternoons at the electrical engineering lab. We went every Saturday for 12 or 15 weeks or something.

Brock: At Virginia you said that the high-level languages really had an appeal to you, and ALGOL had an appeal to you. Could you just say more about that? If you can, characterize what was it about them that was intriguing or interesting?

Austin: I liked the kind of interesting world of problem solving tools that the high-level languages brought. I thought even in college that the Burroughs approach to building computers, which was specifically to make high-level languages effective, because the standard criticism at the time of high-level languages like FORTRAN and Cobol, the two best known at that time, was that, well, sure, they're great for writing programs, but they're so inefficient because you have to translate them somehow or compile them into assembly language, and by the time that happens they're going to be much slower, and they're going to take a lot more memory. You can really do a much better job by just skipping that whole step and writing it in assembly language yourself. Burroughs had this idea that that's not really true if you design the computer specifically for executing high-level language, because then the compile steps are very simple, and you don't lose a lot of efficiency in the compilation. I think they proved that was true. They weren't the only ones that had this idea, but as a company they were the only people that did it.

Brock: If you'll forgive my ignorance about that, I think that's a fascinating idea, and a fascinating approach, the Burroughs approach, or perhaps it has some more generic name. Am I right in thinking that it's down to the design of the logic circuitry would even be involved of the computer to accomplish this, to go kind of more directly from the high-level language to the gates?

Austin: It is. It was really the design of the instruction set that the computer actually ran, and the instruction set was chosen so that you could map very quickly from the higher-level language concepts into the instruction set. Some of the instructions were quite complicated, so they did introduce difficulties for building the hardware, but in many cases the instructions were simple. ALGOL had these problems of procedure calls with passing data to a procedure not just as data, but you could pass a procedure to a procedure. It was called "call by name" in ALGOL, but that's what it really amounted to. This actually was used in other languages later, but the Burroughs machine had a mechanism for doing that where you could pass a procedure as a piece of data, and then the procedure that was called could look at its parameter and see it was a procedure. Actually, it didn't even need to look at it. It could just load it as though it was a piece of data and it would launch the procedure that calculated the piece of data.

Brock: Interesting.

Austin: Yeah, and that was all in the hardware, so it was a very interesting machine. It's unfortunate that the museum does not have a B5500, because it was a groundbreaking machine. It had such things in the operating system as virtual memory, which was essentially an unknown concept in 1960. It wasn't really the first machine to have it, but it was the first commercially successful machine to have it. Gee, what else? I don't know. It was pretty advanced for that time.

Brock: Just to return to one thing that you said that struck me which was about kind of the world of problem solving that the high-level language provided. I just think it's almost, to me, in what I'm hearing -- it's almost like providing an interesting domain in which to think.

Austin: Exactly right.

Brock: Yeah, and did you find that in your experience does that kind of thinking domain and thinking kind of process, thinking process, space of thinking process change from language to language? They have -- high-level language to high-level language, they have a different character?

Austin: Definitely, definitely, especially back then. I mean, I think the concepts in languages have converged over the decades in many ways, but in many ways new things have come along as well. What actually got my interest after leaving college was the idea of designing those domains, designing the languages became very interesting to me, and I did some of that work in the '70s and built compilers.

Brock: Just one or two other questions about Virginia before moving on. One would be just if you could describe what it was like to be a computer user at the university. As a practical matter, how were you interacting with the machines?

Austin: There was three ways to interact with the machine. The way most people did it, especially if you were an engineer and you were just trying to solve problems in electrical engineering you're writing things to do circuits or something. You would write your program typically on paper, and then punch it into cards at a keypunch, and there was a big room full of keypunches. You would go and wait for one to become free and sit down and punch all your cards, and make a deck with your program in it. Then you'd take it over to the desk at the front of the computer. The computer itself was behind a wall of glass. You could look at it, but you couldn't touch it. You'd put your deck in the input tray, and then it was often the next day before you got the output. You'd get your deck back with whatever came out of the printer. Sometimes it was the same day, really, and I can't even remember how we predicted when it would be done. I think they might have had a posting that said about what the typical turnaround was right now, so you could expect when your job would come back out. So that was the main way.

The second way is you could hang around the computer center at night where there were very few people around, and then you could get in past the glass and get to know the operators. Then you could manipulate the equipment directly yourself including the main console, which on a Burroughs computer was called a SPO for the Supervisory Printer Output, I think. It was a sort of an electric typewriter kind of interface, and most of the commands you could put on cards. You couldn't put a program in that way, but you could put the commands that you put on cards like move this file to tape or things like that. Instead of putting them on cards you could just type them at this console. So in that sense it was more like modern computers, or at least a little more modern computers where it was somewhat interactive from the console, and it had a very simple language unlike a lot of other computers. So you could do it from your head. You didn't have to read in a book how to say it each time.

The third way was through teletype machines which were essentially an early timesharing kind of setup where a number of teletypes were available if you had the special card to get into to use the teletypes. In there you could input a program on the teletype. You could put it into paper tape. Each teletype had a little paper tape writer and reader so you could save your work. Those were oriented around the BASIC language system, but you could also do ALGOL from the terminals on the Burroughs system.

Brock: Did you do all three of those practices?

Austin: Oh, yes. Yeah, I did all three.

Brock: Was there a pretty tight community of people who were interested in computers and computing?

Austin: Yes. There was no undergraduate computer science degree at Virginia at that time, and the graduate degree was in the math department, and I'm not exactly sure how the curriculum worked, but they didn't really emphasize using the computer that much, I think. It was more pure study of computer science.

Brock: I just thought the last question about that would be just your reflections on what was happening at the world exciting period of history, the second half of the 1960s. How did that affect your life, what was going on in those days?

Austin: Well, the college campuses were interesting places in those times, but when you're in the middle of it, it doesn't really seem that different from any other time. There was no big disturbance in Virginia during the time I was there. We would read about them in the paper, and there were groups on the campus advocating for various things, antiwar or whatever. I would say I just wasn't that heavily involved in it. I did like to hang out at the coffee house and listen to and play music. So I was kind of with that sort of hippie crowd there, but the rest of the time I was mostly in the engineering.

Brock: As you're coming to the end of your studies, how did you come up with what to do next?

Austin: Well, I did kind of the usual. The companies had come to interview on campus, and kind of get an idea of who's available. I talked to a number of companies. My goal from graduation was to go west in the country, so my goal was to go to California, and I needed help. I needed to have a company hire me in the west, and, with any luck, pay some expenses to help get me there if that would help. I didn't have very high expenses, but at least pay to take me out for an interview because flying out there was a big expense back then. This was before deregulation.

The only one I found that would actually take me west was General Electric whose computer's division was in Phoenix, Arizona. So I ran out of luck trying to get to California. Burroughs was in California, and I knew about Burroughs' computers from the B5500, so they interested me, and I talked to them, but they were only interested in sending me to Philadelphia because that was close by, and that was where the recruiters came from. So I didn't want to go to Burroughs, so I ended up going to Phoenix, Arizona, which was another cultural world away. I very much took to the desert, and I really liked it there. But, as I said, then the company was acquired by Honeywell, and in two years I was in Boston, so I was further east than I started.

Brock: General Electric at this time is making large systems and smaller timesharing systems?

Austin: Yeah, they had three lines of systems. They were basically in the business of emulating the designs of IBM. So they had a series that emulated the 1401, and by emulate I don't mean -- They were just imitating I guess is a better word than emulating. They had a 1401, and then a middle series. I don't know what that corresponded to at IBM. Then they had a 600 series that was like the IBM 7090 in instruction set. So all these were built in Phoenix, and the 600 series, although it had a conventional IBM architecture, they had done some interesting things with it including a language that I was interested in that they called I language. I language was supposed to be used for doing implementation, hence the "I," of software, specifically for the GE 600. I thought that language was interesting. It was not in wide use yet, but luckily after the training program was over I got stationed at the Advance Systems Technology operation, which was a small building outside of the main factory in Phoenix, and that was a delightful place to work, and that was where the guys that built I language worked, and there was a new machine being designed there. For that they were doing an improved version of I language, which was going to be called the Q language, and I was assigned to work on the Q language compiler, which was being written in I language. So it was a much different place than I thought it was going to be. I really hit the jackpot as far as my own interests and so forth.

Brock: When they made you the offer did they kind of give you an idea of like we'd like you to work in this type of an area, or was it really come work on computers?

Austin: No, it was, "Right, you're a new undergraduate, or new graduate from undergraduate school, and yeah, come and we'll figure it out," because very few people had computer science degrees, and you just had to have interest and they'd see where it went. In fact I started off in a training program kind of thing there that lasted maybe two months where you went and visited different areas of the company, and met people, and there were a lot of lectures explaining what GE's products did, and it was a great way to start.

Brock: It sounds like in your case it really worked well to make a match of your interests.

Austin: Yeah, it did, yeah.

Brock: Maybe you could just describe what it was like to come up to speed on that new language, I language, and to work on the compiler. What was that work like? How did you do it?

Austin: Well, it was kind of the punch the cards and submit them at that the desk, although we did have access to the real computer, and we had a 600 right there in the building even though we only had about 35 people there. I quickly learned how compilers worked. Incidentally, a number of the people that had designed I language and designed Q language had Burroughs in their past and had ALGOL in their past, and it was reflected here, and there was even some concepts from ALGOL 68, which was a language that never really went anywhere, but had a lot of new concepts that were worth examining. The cross-fertilization was quite surprising from Burroughs, actually, and the new machine they were designing had a lot of Burroughs influence in it too. The guy that I learned what Q language was like actually was a previous Burroughs guy who was independent consultant and teaching at the time, Dr. Dave Dahm. He eventually went back to Burroughs when I was there later in my career, and I got to know him pretty well. He was a great guy. So you find out where these people came from. Many of them came from Rice University, Dave Dahm did, and several other people that were there. Rice made a computer that had some relationship to Burroughs computers, shall we say. It was a stack designed to their architecture. So, all these ideas sort of get traded around as people move from place to place.

Brock: Would you say that's one of the main way that these concepts and techniques were being transposed and transmitted in those days as people moved around?

Austin: Definitely.

Brock: Yeah?

Austin: Yeah.

Brock: Did that change over time, or was that still throughout your career was that the primary way that approaches and techniques -- they moved with the people?

Austin: Yeah, it's probably that way in most industries. The more cross-fertilization there is the more ideas run from people wanting to do a better version of something they did in the past. Now they know their mistakes, but the company where they first did it no longer wants them to work on it because the first was a failure, and so they need somebody else.

Brock: I guess you went from working for an organization as part of a very large corporation to then with Honeywell, a slightly less, but kind of also very large organization. How did you find that, working in the context of a large company?

Austin: Well, I sort of got into the same niche at Honeywell, because this group that I mentioned that I was with in Arizona with GE, the whole group got moved to Boston. So I went with the group. Its mission changed, and we were no longer programming in I language, and we were no longer building Q language. Instead there was a special version of the GE 600, which was at MIT. There was actually two of these built, one at MIT and one at Bell Labs in New Jersey. It was the 640, and this had some special hardware in it for doing virtual memory, and a few other things, and was the basis for the MULTICS operating system that was a cooperative project between MIT and Bell Labs. When I located to Boston I was outside of Boston and Waltham, but it was just a 20-minute ride into MIT, and we started using the MIT MULTICS computer for doing our work. Eventually we got another 640 out locally in Waltham, and so then I became a PL/1 programmer, at least the special version of PL/1 that was used for MULTICS. The compiler I started working on then instead of Q language was a new PL/1 version for this new machine. As I said, I was going into classes at MIT, so I would go in there and talk to the people at the lab in there too.

Brock: Honeywell's operation in Waltham- - they had acquired a company there, wasn't it? Was that the origin of the Honeywell operation?

Austin: I don't know too much about Honeywell's before. They acquired GE, of course, but that operation all stayed out in Phoenix. I know Billerica and Waltham were the two places where programming was going on, and I think they built computers in both of those locations, but I don't really remember how they got to that stage. Honeywell also was an imitator. Most other computer companies imitated IBM designs, not CDC, not Burroughs, but most others did. RCA was in that business.

Brock: For how long were you working with that group for Honeywell?

Austin: For Honeywell I was there until '73, so I guess it was only about two and a half years, and I resumed my quest to move west. Now I had accumulated some more stuff, and a wife and her stuff, and now it's going to be a little harder to get west, so I was trying to find a company that was in the west that was interested in hiring me. I went to quite a few looking trying to get interviews, and I got a couple, but I couldn't get one in Burroughs still. If you didn't start off in the west they weren't interested in bringing you out there. I don't know why they didn't seem to be interested. So I went out for an interview with NCR, which was another imitator. They're in San Diego, Rancho Bernardo. I called the Burroughs personnel department in Santa Barbara, which was a place I was interested in going and said, "I'm going to be in San Diego next week. Would you be interested in talking to me? I could just drive up," and they said, "Oh, sure." So I got an offer from both NCR and Burroughs in Santa Barbara and went to Santa Barbara.

Brock: I would imagine at this stage you more clearly knew the area in which you were going to work.

Austin: Yes, I did, yeah.

Brock: How would you characterize that? When you were talking to the people at Burroughs how did you describe what you wanted to do?

Austin: Well, I was, by this time, established as kind of a compiler writer, and they needed compilers, and I was particularly interested in the kinds of compilers they were doing because Burroughs was the small systems office, and they had a very unusual machine. All of Burroughs' machines were unusual. I described a little bit about the 5500 before, which was built to execute ALGOL, and they had a medium sized range, the 3700/4700, which was designed to execute COBOL, and was their real business line, and then the small range, the B1700 was the most interesting of all because it was their first micro-programmable machine. Their idea was unlike other micro-programmable machines they would change the microprogramming depending on what language was being executed. So every compiler created assembly code, basically, for a different machine, a different virtual machine. So the COBOL had its own virtual machine. FORTRAN had its own virtual machine, and when the operating system ran a FORTRAN program it would load the FORTRAN virtual machine into the microprogram memory.

Brock: Could we talk about that? It's just so I can make sure I'm really following, but if you don't mind. As I understand it, is the microprogramming kind of a layer that stands right above the instruction set, and kind of defines. It's kind of one layer of abstraction up from that. Maybe you could explain to me.

Austin: Well, it is, but in this case the instruction set itself, the hardware instruction is extremely simple.

Brock: Okay, so it has very basic…

Austin: It gets stuff from memory and that's practically it. You almost have to teach it how to add. It did have an adder in hardware, but no other operations. So the idea was you could write this interpreter, which is what the microprograms were called that did this, so it would look at the instructions that the compiler generated, and then execute the microcode to affect those instructions. So it made the hardware simpler, by basically building part of what used to be in hardware into a new layer of software. This idea of interpreters for computers was reused even without microprogramming later quite a bit.

Brock: So for the Burroughs small systems that they were making in Santa Barbara, it had, if I'm understanding, it had different microcode associated with the different high-level language.

Austin: Exactly, right.

Brock: This was a real different approach from others?

Austin: I know of no other machine that worked this way.

Brock: Okay, so highly unusual.

Austin: Yeah, it was very unusual, and it was a great design space to work in because we could not only design the compilers, but we could design this sort of intermediate layer to try to make things optimal. You could make something easy for a compiler to generate, but easy for the interpreter to understand and execute as well. So you had sort of two layers to optimize there.

Brock: In terms of the business success, I guess, was that a popular line of computers?

Austin: It did fine. It kind of fell by the wayside in speed eventually, so the approach we had we couldn't really make it fast enough with the style of approach we used. So it went through a few generations B1700, 1800, 1900 and so forth. In fact, the museum does have a 1900. It just became obvious with the approach we were using we couldn’t keep up with the speed. That was at a time when Burroughs ended up getting acquired by Sperry Rand. The 1700 was the line they decided to end. They continued with the other two lines for a while, and eventually they only kept the large systems going.

Brock: When did that merger/acquisition happen?

Austin: That was ’81, I believe.

Brock: Were you still there at that time?

Austin: I was still in Santa Barbara at the time of the merger.

Brock: You must have left not soon after that.

Austin: Yeah, they decided to stop new development, basically. The Santa Barbara plant went on to make computers for a while, and to maintain the old stuff, so you could stay if that’s what you wanted to do, and if you just loved Santa Barbara, and believe me we all did. I wanted to do something else, so I moved to the large systems and took a crack at that for a while.

Actually, my job at that point was trying to see if we could implement UNIX for the large system’s architecture. That seemed kind of an unlikely project to me. I knew C by this time from college courses and stuff, so I wrote a C compiler for the large systems, and started to port stuff over, and began to realize that it was just not going to be a successful project. In this case the computer itself was sort of higher level than the higher-level language that we were using. It was a very odd sort of thing to try to implement. The hardware really had lots of things in the instruction set and you couldn’t even write in C. So I kind of killed the project myself. I wrote a long report explaining why I didn’t think this was going to be economically feasible. It would be feasible, but it was pointless. So that was the end of my time at Burroughs, and I moved up to Silicon Valley.

Brock: A couple of questions about those Santa Barbara years. For people who are interested in compilers and high-level languages, what was the kind of professional community like in terms of people nationwide or internationally? Were there meetings that you would go to regularly, or particular things that everybody seemed to be reading? What was just that community aspect like?

Austin: Well, Burroughs had hired a lot of very young people to start this plant in Santa Barbara in about 1969 and ’70, I think. So there’s a lot of young people. There was a very collegiate sort of atmosphere, and a lot of smart people. We had UCSB, and as I mentioned I learned C and UNIX taking classes over there, so I sort of had a parallel education going on. In addition, there was a well-known character named Edsger W. Dijkstra who was writing books and papers, and so forth about programming, who was hired by Burroughs as a consultant at the time. So Dijkstra would occasionally visit and give a talk or something, and he was a very eccentric and fascinating person with very provocative views, and some people liked them and some people didn't. I just thought it was interesting he was so provocative because, again, I like these odd things. I got to know him a little bit through his visits, and we corresponded. I would give him a problem, and how I'd solve it. It was very interesting.

Working with the other plants there was also some cross-fertilization. I built a Pascal compiler for the 1700 while I was there. That was just sort of a night and weekend project, really. So we had Pascal running, and I tried to get other people to write Pascal stuff for the 1700. I found out there were people back in the Philadelphia area who were interested in Pascal too, and we decided that Burroughs needed a standard Pascal compiler, and so I started working with them. We had a Burroughs committee on standard Pascal for that. So that was another sort of group thing. Then I went to the Pascal Standards Committee for the larger standard as well.

Brock: Who convened that?

Austin: I can't remember who convened the larger one. It certainly had nothing to do with Klaus Wirth, who was in Switzerland at the time, but he was around and writing for sure. There was a group at San Diego that had created a Pascal that ran on personal computers, well, Apple IIs anyway, that was very interesting. That was an interpretive system, by the way. It's similar to the way the microcode worked on the 1800 except it wasn't microcode. It was interpreted and ran directly on a microprocessor on an 8080, or a 6080.

Brock: Right. During your time in Santa Barbara was also the period of this first emergence of personal computers. Could you talk about your growing awareness of that, and your thoughts about it, and maybe what other people around you were making of that development?

Austin: Well, I would say we weren't convinced that personal computers were anything that we were interested in intellectually. They were fascinating in that you could make computers this cheap and that you could play with them at home if you're willing to invest a few thousand dollars, but they didn't seem intellectually interesting. In fact, it seemed that the personal computer software was determined to repeat every error that had already been made in mainframe software <laughs> and they--

Brock: In what way?

Austin: Because they had no discipline in the way they wrote software because the machines were slow and had very small memories. The same kind of situations in early mainframes and so they just sort of hacked away in assembly language to create interesting little things and that was very much the kind of toy atmosphere there was in early mainframes, but by now we had discipline. We had high languages. We had operating systems. We had a lot of stuff to work from that wasn't around on PCs and yes, the hardware was kind of interesting to be able to play with, but it just didn't seem like a place I professionally wanted to work.

Brock: Certainly given your interests, none of that was happening.

Austin: The thing that changed my mind was VisiCalc.

Brock: Really?

Austin: Yeah, because when I first saw VisiCalc, went to visit a place at Burroughs that had bought an Apple II and they were experimenting things and the immediacy of VisiCalc was very persuasive because it was the first time I'd seen a program with that instant interaction with a user and it was much different than the idea of typing commands and making it do something. This was type a number and watch it propagate through a spreadsheet. That was interesting stuff to me. <laughs> Yeah. That was. I don't know how. I think these personal computers would be something very interesting.

Brock: Well, I guess as things were changing in Santa Barbara and with Burroughs and you started I guess thinking about what you should best do, how did you come across what to do next and to come up to the San Francisco Bay area?

Austin: Well, at the time I had moved around the country quite a few times and managed to stay in Santa Barbara for about nine years which was delightful, but I had to move again as Santa Barbara didn't have much choice and the Bay area, and specifically Silicon Valley, was the place you went when you never wanted to move again because there were so many companies here you could always find something interesting and so I decided to come up here. I knew quite a few people here from previous jobs who had ended up migrating here, and I came to work for Tandem which had quite a few people from Burroughs there, and again I was working on the design of a new machine there and a new language. Tandem just didn't work for me very well and I decided it was time for me to move to the PC world.

Brock: Tandem just because of the project or the culture? The circumstances?

Austin: It was just kind of the group and it was still a mainframe project. I was doubtful about the way the project was being carried on. It was a second system kind of project where Tandem had had a very successful system and now they were redesigning. They had enough money and talent to really do it right the next time. Unfortunately they burned untold amounts of money <laughs> and talent trying to do it a little too good, which is a syndrome that a lot of companies fall into. They make the next step too big. <laughs>

Brock: That's a recognized--

Austin: Yeah, it's called the second system syndrome.

<laughter>

Brock: I've not heard that before.

Austin: And so I left to go to work for a portable computer company, Gavilan Computers, or, as they liked to refer to it, a mobile computer company. It always seemed a little weird because they didn't have wheels. It didn't seem quite right to be mobile, but they liked that better than portable because Compaq had a computer that was about the size of a carry-on suitcase that was called the portable and--

Brock: They wanted to distinguish.

Austin: Yeah, and they didn't want to use that term, so they used the mobile computing and the Gavilan computer was a laptop that was battery operated. It was the first battery operated laptop computer. There was one kind of well known one, the Grid computer that had to plug in that predated the Gavilan, but we were completely self contained and interestingly again it wasn't just going to be another computer. <laughs> We were going to design a graphics user interface for this little computer. We were very ambitious. The computer just had a little tiny LCD black and white dot matrix display and so you couldn't do a lot of graphics on it, but you could do graphics and we had a-- it wasn't a touch screen, but we had a touchpad below the screen that you could drag your finger around on like the touchpads on a modern laptop and you could tap on the pad and it had a few other touchpad buttons there.

Brock: To move a--

Austin: To move an arrow cursor around this little tiny graphic screen. It was very ambitious.

Brock: And two things. Had you -- by this time you had encountered the graphic user interface and one, if you could just talk about that. I mean we touched on a little bit in our conversation yesterday, but it might be nice to talk about it again in this context, and also it sounds like this is your first experience of a startup company.

Austin: Yes it was.

Brock: Maybe we could talk about those two a little bit.

Austin: Okay. Yes, I was already sold on the graphics user interface idea from a friend from Burroughs who had gone on to work at Xerox PARC, Wayne Wilner. When he was at PARC, he invited me over for a demo on how the Alto works, so this was just in his office. It was not an official demo or anything, but I spent an afternoon there and became pretty entranced with it. Of course I'd read about it before so I kind of knew what to expect, but it was very entrancing to see it all work and see how the word processor worked and servers and things on their network and I had followed a lot of the activity partly because of the languages I was compiling and they had also created their own language and architecture for executing that language, the Mesa language. So the whole thing was fascinating to me all the way from the design of the hardware up through the mouse and screen.

So that was my first experience, and at my next experience I knew somewhat. I studied how to do it. We had to write these things to move bits around (the bitblt operator) and so forth on this little Gavilan computer and we bought a Lisa at Gavilan to use to kind of -- for people to get experience in how these things work and so I spent a lot of hours messing around with a Lisa understanding how graphics user interfaces work at that time, tried out all the applications and that was very instructive. So that was my kind of path into that area.

Gavilan itself was an interesting company. The guy who was the director of engineering is still a close friend of mine from those days and he's now at Google, John Banning, and there was really, people form -- we were in a Zilog building down in Campbell and there were some others from Zilog there and the implementation was all done on a Zilog Z8000 Unix computer that we would then put the software on the portable from there and the president, Manny Fernandez, was a Cuban guy that was just wonderful. I mean he was just a terrific leader for the company and John Banning was just a great engineering guy and so I had a very good experience there, really good people I worked with.

The company was managed really on the hair edge the whole time and managed to spend about $300 million as I recall in two and a half years or something amazingly short. <laughs> I may have the numbers wrong, but the company was, as they called it, spring loaded for success. They had so much invested in it. All it took was a little income and once it started to grow, they were right with it. They had everything in place, but unfortunately it didn't grow fast enough and the software we were working on wasn't really ready and all they had was they could run MS-DOS on it, so it could fill that need, but our really nifty graphics user interface was going to have to hold back a few more months and eventually the company just had a spectacular bankruptcy. So that was the end of Gavilan.

Brock: And what was your role and your title there?

Austin: I was something like director of architecture but I never thought that described what I did very well. I called it architectural coordinator because I went around to people working on pieces of it and made sure they were working together was actually a lot of what I did. For the code I wrote was all user interface code so I was always kind of interested in user interfaces and this was the first time I had actually done a lot of extensive coding on one. When I talked about the thing I liked about high level languages as being this model of how you understand things, I always looked at user interfaces kind of the same way except it's an interactive user interface rather than one that you are writing and describing problems in, but I saw the same thing attracted me to both ideas.

Brock: I could see that, how it kind of defines a space. It defines kind of a world, as you said, which you're operating.

Austin: Exactly.

Brock: The metaphors and yeah, it is interesting to think about that structuring kind of a way of thought.

Austin: That's exactly what it does, yeah. Yeah.

Brock: Well, so how did people let you know that it was spring loaded for success but the spring never gets <laughs> sprung? So was everybody talking about the possibility that or how did you know that things were ending badly?

Austin: No, it came pretty suddenly. I mean we could see that sales were not there, but we really didn't understand the finances of the company because it was by that time it had a couple hundred employees. It looked like things were going well because they had a lot of financing obviously, but and we had interesting press and showing the machines at computer conferences and so forth. We had a very polished image, but it sort of just all collapsed within a few weeks.

Brock: Wow.

Austin: Yeah. "We don't have any more money."

<laughter>

Austin: It was pretty sad, but -- and by that time of course the engineering team as often happens was really a well-oiled machine that really enjoyed working together, understood each other, and it seems such a shame to tear such a machine apart and we thought it would be great if somebody could just hire the whole department. We were just ready to go, but of course that can't ever happen, so we wandered off in different directions and there were a few reunions over the years, but--

Brock: Well, what about for you personally? I guess the region was proving itself in your case that you wouldn't have to move again.

Austin: Exactly.

Brock: So how did you? Let's see. So what year? That would've been 19--

Austin: This was '84.

Brock: Yeah. Was this an uptime or a downtime in Silicon Valley at that?

Austin: Oh, I think it was a little slow in the financing area, so it wasn't like a lot of companies had been founded, but there was not a lot of new money available. It was kind of going on and so going to a new startup you needed somebody that already had financing and I wanted to continue to work on user interfaces I thought or applications design, the user interface for an application, and I found this company Forethought which I'd read a couple things about in magazines.

It hadn't been much written about yet because they really hadn't done anything much, but I saw an article that said that they'd hired Bob Gaskins, whom I knew, and I thought well that sounds interesting. They were financed by the same venture company that financed Gavilan which was New Enterprise Associates and New Enterprise Associates I was kind of impressed with in their Gavilan days. I mean even though we were out of money and going bankrupt and they were losing their investment, they threw a party for all the employees in the company as far as to say goodbye and sorry it worked out this way at their own expense. So that certainly continued to leave me with a good impression of NEA and the lead guy from NEA who was working with Gavilan was C. Woody Ray and I had met him a couple of times and I spoke to him about Forethought asking, "So what do you think about this company?" Because he must know all about it if NEA's financing it and he said, "Well, they looked promising, but I don't know. They haven't produced much and I don't think it's that likely to succeed, to tell you the truth. So maybe that's not the place for you," since I'd just come from one failing company. So that put me off a little bit, but I was still kind of sold on Forethought's ideas of what they wanted to do. They wanted to create a system on an IBM PC that would be like a Lisa.

Brock: And how had you known? I can't recall how you had known Bob Gaskins previously.

Austin: Well, I had known some people from Burroughs who had gone to work for Bell Northern Research where Bob worked. Some of my friends worked in Bob's group even. Some were in other parts of it and that's where Tom Rudkin met Bob too. I think I met Bob a little earlier than that, but it was very close to the same time. There was a conference given at the University of Santa Cruz when I was still at Burroughs that I attended. Yes, I was still at Burroughs because I was driving up. It was a two-week conference with pretty much all my heroes of computer science at the time and I got a chance. You stayed in the dorms and it was a very intensive sort of event and among them Edsger Dijkstra, Tony Hoare, and Niklaus Wirth and many others that I knew of that were doing work in proper ways to program, languages, discipline in programming was Dijkstra's theme, and structured programming. That term was probably already going out of fashion by that time, but that was how it started. That's where I met Bob by the way, is how that started.

Brock: Yes, yes. So then did you just put in a phone call to him or something after you became interested or did you?

Austin: I don't actually remember. That's probably likely. We didn't send email in those days. I mean we used email because anybody using a Unix system, the users on it communicate in email on that system, but we didn't have an internet. So you didn't send it outside the company.

Brock: And how did -- when you initially were talking to the folks at Forethought, were they, the vision they were still actively pursuing was this turn an IBM PC into an Alto or something?

Austin: That was, yeah, the official plan. Because of the fact that it looked like their income from that was going to be farther in the future than they thought, they tried to raise some money themselves by going into the Macintosh software publishing business and they found a few authors who had programs that were in need of a publisher and so they had this line called Macware.

The first Macware product which was called Fact Finder, and it didn't last very long, but that shipped on the day I started. So they were just starting into that line and there were several products in the line. They had planned about five, but they had three or four, but their success story was FileMaker which is still a product. When I started there, everybody in the company helped with that activity as well. It was a very small company. It was about 12 people, so--

Brock: Where was it physically located?

Austin: It was within walking distance of the museum here really, on the other side of where the theaters are, over there. The buildings used to have blue roofs and bright blue. They were very distinctive from the freeway, but they're painted some other color now so you don't notice them.

Brock: Just kind of over here?

Austin: If you're pointing the right way. Just up along the freeway toward the north.

Brock: And let's see. Well, it seems that Rob Campbell had done a similar sort of software publishing activity while he worked for Apple Computer. Was that part of his job there? I can't recall.

Austin: I think his job there, he had a small company that made accounting software that ran on Apple IIs and he moved from there to Apple and I think he was more of an evangelist kind of thing or what later became known as an evangelist at Apple where he would work with companies to help them market and design stuff for Apple computers.

Brock: I see. So this move to the actual software publishing was not the--

Austin: Yeah, right, and he certainly understood the business and software publishing was-- I don't know. It was a business model that was fairly prevalent at the time and then it just kind of evaporated. I'm not exactly sure why that happened, but now it certainly wouldn't make sense because the smallest authors can get published by just publishing on the internet so you don't really need a publisher. You don't need printed manuals. You don't need a lot of the things you needed back then.

Brock: Well, what was your first order of business when you joined Forethought?

Austin: Well, I thought the very first thing I wanted to do is understand the Macintosh which had been shipped from Apple about six months earlier. I knew about the machine. I mentioned I used the Lisa, but the Macware products all ran on Macintoshes and I'd never used one, so that was the first order of business to understand how that worked. As a guide to stuff we would do in the future, I needed to understand what the market was and what the applications were for these graphics style machines.

So that didn't take long, and then I needed to understand what Microsoft Windows was doing because Bob's idea was to take the work that had previously been done at Forethought which was a complete system from the hardware up to the fonts <laughs>, operating system and everything, so he thought that Microsoft Windows would be the way of the future and Microsoft Windows didn't even exist yet. It had been announced, but it wasn't shipping, and his idea was to take whatever application level stuff Forethought had and get it to run on Windows because that would be the better way to do it and would be more standard. We would have less effort because we could leverage all of Microsoft's work. So that's what my initial job was, to try to get their existing stuff onto Microsoft Windows. The staff that had created the existing stuff was mostly depleted when I was hired. They had all been fired except for one because money was leaking at a rapid rate and they could see that things didn't look successful. So one guy was left, Peter Bishop, who was a very smart guy and had come from Xerox and he basically understood what had been done so far and the two of us worked on that and tried to get it running on Microsoft Windows. So we spent a few months on that and we had a demo day when we were supposed to show how this stuff would work on Microsoft Windows so people would understand what we had. We ended up canceling the demo day <laughs> because we showed it to Bob Gaskins, because he was our boss, and Bob thought we can't show this to the investors. It's not going to go well.

<laughter>

Austin: I agreed. It was not going to go well. There wasn't much there and so we ended up canceling the whole idea of the stuff that had been done so far which was a big cost, but there was just not enough there.

Brock: Was that? Maybe we could just touch a little bit on what that task was like. So what the original Forethought applications, were they written in a language and in a way that made it easier or harder to modify so it would work with Windows?

Austin: They were written by people that really wanted to build an extremely strong foundation. It was written in C and they had, but a lot of it was very low level operating system style stuff that all went away when we went to DOS and Windows. So we didn't need much of what was already done. So it's mostly the application level stuff which they hadn't spent as much time on. There really wasn't that much yet as far as apps. There was a little graphics engine, and we showed how you could use it, and it ran on Windows when we ported it over, but they had been building a very deep foundation and hadn't really been working up at the application level much yet, so when we moved to Windows, there really wasn't much there. <laughs>

Brock: And then okay. Did developing for DOS and Windows, did that require using a different programming language or could you still use C?

Austin: No, this was all in C. Yeah, we started using the C that Microsoft sold to do this in. Microsoft's had an interesting approach in that they used C internally too, but none of the software that Microsoft produced was actually written in the C compiler that they sold. <laughs> I found out later, in working for Microsoft, that the stuff that the C compiler group made, they sort of disdained as too big and slow, and it didn't suit the applications group, so <laughs> that didn't really work for us because we had to use the one that they sold. It took quite a few years after I was at Microsoft to get these two C's combined. The applications group, like any applications group, they're selling their stuff and it's got to look good, and it's got to look fast, and they just didn't think that C compiler was going to cut it. They wanted control of the compiler so they had their own.

Brock: Interesting. Well, so after the canceling of the demo day, does that represent the time when really you and Bob Gaskins began talking about presentation software as something to do next?

Austin: We started looking at ideas and one of them that came up was one that Bob had proposed the previous year to the other people, which up until that point I had not heard about, which was presentation software. I was very interested in typography and graphics in computers and that kind of thing, but presentations as a business was not something I was really even aware of. I'd been to a lot of presentations, but that didn't seem like a category to me. It wasn't really much of a known category. It was a little bit there. I was more interested in sort of the page layout kind of things because I was certainly aware of the printing industry and using these kind of computers for that. It's something that Xerox had pushed from the beginning, so I liked that whole idea, but Bob's idea of doing it for presentations, when he brought that back out as a candidate for where we ought to go, that sounded yeah, that this could work. It's sort of a new category. We wouldn't be competing with people in the publishing business on a page layout business.

Brock: Right.

Austin: PageMaker was not out yet by that time, but PageMaker became the standard, Aldus PageMaker which eventually went to Adobe and yeah.

Brock: Right, and that must've come out in this era.

Austin: Yeah, that was probably '85. Yeah, something like that. At least I think it was the guy from Aldus, the company that originally made PageMaker, Paul Brainerd I believe was his name, he showed it at Forethought because Forethought was interested, or was a Macintosh publisher, and at first his idea was to get it published by Forethought and a deal never was forged there, but we were aware of the product.

Brock: That is an interesting aspect of having that software publishing side is that you're very current on what other people are producing.

Austin: Exactly. Yeah.

Brock: Well, let's see. I mean I guess we don't. I'm just thinking of the time we have left and our discussion from yesterday where we with Tom discussed the process for well, imagining PowerPoint, realizing PowerPoint 1, realizing PowerPoint 2. I wonder what do you think? Should we maybe instead of trying to go back through that unless you feel that there's something that we missed or that we might want to explore in more depth, we could talk a little bit or we could talk about two things. One would be kind of your experience with PowerPoint and Microsoft from 1990 until when PowerPoint 2 came out until whenever it was that you left Microsoft and then talk about your experiences after to the present and I would also then maybe close by some time just talking about your thoughts about just along the way the sort of reflections on what makes for traits that make somebody good at programming, just the experience of creating software. Just reflections of software that--

Austin: That's a lot of ground, but <laughs> potential--

Brock: Yeah, well maybe we could try. Does that sound like--

Austin: Well, let me start with the imagining PowerPoint just a little bit more--

Brock: Yes.

Austin: --because that was one of the most interesting times in my career because Bob had this one-page description of what the software ought to do for presenters and so I pretty much had the freedom to figure out how that would work in a program, and it was the only time when I've had that much latitude and just kind of starting from scratch. It would be a lot like designing something if you didn't even work at a company. You were just doing it in your basement except that I had Bob who is a literate and smart guy with a lot of good judgment.

The process I went through, I would come up with ideas, and I would draw some pictures and I would write some text about how things would work until I had an idea in my mind, and then I'd go in and explain it to him for a couple hours. He was always available. We spent a lot of time together and we would talk it over, and the process of explaining it always made it more clear to me to try to make it understandable to him, and then he always had some good comments and why something was better than something else, and then I'd go back and we'd cycle through this again. <laughs> It was a very successful process because we were both so interested in its success and we didn't have fighting ideas. I was sort of the idea generator and he would take the ideas and see whether they fit his <laughs> thought about what the program would be eventually and in the course of a couple of months, we pretty much knew what PowerPoint was going to be like. So that was a very special time for me <laughs> coming up with all that.

Brock: Were there other people that the two of you brought into the dialog?

Austin: No, there was nobody else there really that really had the interest in this kind of fine detail. I was the only programmer and the other people were more in sales and marketing and that kind. Anyhow, yeah. That's just the one thing I wanted to add about the beginning of what you were saying.

Brock: Right. Would you say that that was I guess that sounds like it was a unique process for you then.

Austin: It was. It was a very successful kind of collaboration.

Brock: Did you try and repeat the form of that in later endeavors?

Austin: I think we did. Yeah, I think we did a lot. I remained the primary designer of PowerPoint through PowerPoint 3 at least. I wrote all the specs myself and sort of put the ideas together. After PowerPoint 1, even by the time Tom was there, he was contributing good ideas too. The more people you have, the more good ideas you have, but I was always the coordinator up through PowerPoint 3, I think.

Brock: And PowerPoint 3 was released in 1992.

Austin: Ninety-two, I think, yeah. And that was a big step and in fact it was released first for Windows. All the time PowerPoint 3 was being built, the Windows and Macintosh versions were being developed concurrently so you could try out ideas on both of them, but when it came time to actually buckle down and get the bugs out and get it out the door, it's hard to do both versions at the same time. <laughs>

Brock: Right.

Austin: So we went to the Windows version which we thought was where the money was, so that's the one we got out first and then the Macintosh version we sort of repeated the process on and came about two months later.

Brock: In both your kind of essay about the creation of PowerPoint and also in Bob Gaskin's book on the same topic, I think both of you produce an interesting table of I think person months required to create the different generations of PowerPoint and it seems that there's and the table shows this dramatic increase and I was puzzling over that, quite frankly, puzzling over that table thinking what could it be <laughs> that it went to an order of magnitude, a greater amount of effort, and I was thinking perhaps it's because of the software dependencies and interdependencies involved. I was either thinking the program does a whole lot more or it has to fit with many more things. It has to be more tuned to maybe this ecosystem of software. Which one was it or was it a combination of both or something different entirely?

Austin: Well, I think doing this idea we had of doing Macintosh and Windows versions of the software that was exactly the same and using the same code as much as possible for both of them represented a pretty big drag. There's a lot of overhead in trying to do that. It's much easier to just pay attention to one system at a time. We had for the Windows version we had to kind of bring the Windows along with us because it was not nearly as advanced for graphics applications as Macintosh was, so we were constantly struggling with working with the Windows people to explain to them what we needed to be able to do the same things on Windows that we could do on the Macintosh and that represented a pretty big drag too. Windows was a little harder to work with. We had memory limitations and so forth that we really didn't have on the Macintosh.

The fact that the Windows hardware was more diverse, made by different manufacturers, means you're addressing a wider class of computers and that causes a lot of pain too. So programming for Windows was just harder than programming for Macintosh partly because of the hardware situation and partly because there wasn't enough support in Windows yet and we had to kind of explain what we needed to the Windows people and by the time we got to PowerPoint 3 and Windows 3.1, we pretty much were there. So Windows had what we needed and we understood how to use it. <laughs> But there was a lot of programmer effort in there that was somewhat nonproductive. I mean we couldn't have accomplished our ends without going through those steps, but it was a lot of work.

Brock: As you know that it requires a greater span of time to develop the next version of the software, I would imagine that at the beginning of the effort you have to make some sort of assumptions about the hardware on when it's complete, your targeted imagined hardware, I suppose. I wondered if there was a formal discipline about that or I guess within Microsoft or was there a common kind of target hardware that everybody was imagining or how did that work?

Austin: I think what you say is true a little bit, but I was always afraid of that kind of approach because I was a very practical designer and I was very afraid that things would not be sufficiently interactive and so for designing the original PowerPoint I was very tentative about every idea I had, I would code something and see is this going to be fast enough and I was constantly worried about that and trying to keep the scope of the product that would work on that computer, on those little computers which seem incredibly <laughs> simple and slow by today's standards, but they seemed fast then, but doing graphics interactively is very demanding and something that I wasn't that used to, so I was very suspicious and was always trying everything out and then by the time we shipped PowerPoint 1, of course the Macintosh II was out. It had color and it was twice as fast <laughs> so if we had been able to design for that to begin with, we would've had a much better product in the Macintosh II, but it really wouldn't have been good for the old stuff, so it's often better to design for what you have at the beginning because even though that might not be the advanced hardware when you ship, there's still going to be a lot of people that own that stuff--

Brock: True.

Austin: -- and they're going to be customers too.

Brock: Right. Probably a far larger number of them.

Austin: Yeah.

Brock: That's interesting. Well, after being the lead architect or architectural coordinator, however you choose to describe your role through PowerPoint 3, and I guess that your Forethought had moved over to Palo Alto from Mountain View. Is that correct?

Austin: Yes. It actually moved to Menlo Park.

Brock: Menlo Park, yes, and then were you in Menlo Park then through PowerPoint 3?

Austin: We were there through PowerPoint 4, I believe.

Brock: Okay.

Austin: Yeah.

Brock: And when did you leave Microsoft?

Austin: I left in '96.

Brock: Ninety-six.

Austin: I had taken a leave of absence a little before that at the end of '95 to try to see whether I could find something else in Microsoft I wanted to do and see if I could just have somebody take my place actively and I'd still be around to help. So I had a very capable guy that I thought could take my place and indeed he's worked with PowerPoint. He's still back on that project again. He's done other things in Microsoft since, but he's a really smart guy and so where was I going?

Brock: You were just thinking about is there something else I could do in Microsoft.

Austin: Yeah, there was an advanced technology group in San Francisco that I wanted to work for and they wanted to hire me actually, but they made me interview with the advanced technology people in Redmond and I didn't really pass muster with them for some reason. <laughs> I don't know. I wasn't research-y enough in background I guess and they were real researchers. The group down here was headed by Jim Gray who actually was a early guy at Tandem and a famous database guy, a wonderful guy and I'm sorry I didn't get to work for him because we got to know each other fairly well, went sailing on the Bay. He had a sailboat. Yeah.

Brock: So you obviously decided that you would choose to go.

Austin: Yeah, so at that point I left Microsoft and after a couple of months off I joined a woman…we had a little consultancy and we had an office in Los Altos, the town where I live, and we worked together for ten years there.

Brock: And was it she who developed that iconic clipart?

Austin: Right. One of our activities was selling Screen Beans clipart which she drew. Right, and she's actually fundamentally an artist. She worked in marketing and was quite successful at it and you can make a lot more money than you do as an artist, but she's recently retired full time and now she's a painter. <laughs> She doesn't have to make money. She can spend all her time painting.

Brock: And what sort of activities did you in particular enjoy in that consultancy? Anything stand out for you?

Austin: I enjoyed the freedom. The small company, just partners, we did some interesting consulting projects and I wrote the software for selling the clipart products online and I hadn't done any web programming before that, so that was an interesting experience. I was the tech guy and she was the artist in that part of the business and we had both had some padding from our time at Microsoft, so we didn't really have to work that hard. It was the first time I had worked eight hour days in a long time.

<laughter>

Brock: Well, maybe we could then talk about just some of those reflections on programming and reflections on the sorts of when you were looking for colleagues to work with in developing software, what kind of traits and things, what you would look for and thoughts perhaps just about the place of software in our world and where you may see things going.

Austin: Well, I was heavily influenced by those days when I was at Burroughs and my experience with Dijkstra at that time and his idea of structured programming and discipline in programming and understanding very precisely what you're doing rather than the more modern view in which the word hacking is often used for programming in the last 20 or 30 years. Dijkstra was the anti-hack. <laughs> He used to understand everything mathematically to the last detail. You don't need to test your programs because you can prove they work before you even run it. Well, that was difficult, <laughs> but I was still heavily influenced by that and I've always tried to have a lot of discipline behind my work in programming and I look for that in other people too. So a good programmer is a combination of discipline in writing things carefully and with the kind of structure that'll last, and creativity is the other side of it. What you're creating and the structure of what you're creating is another thing that makes it last. It's not just testing it to death. So you look for that kind of thing in people you work for and with, I think, if you can because that's a compatibility factor. We did pretty well at that, certainly at Forethought when it was just Tom and me and Bob. We all saw it. I know that very well and it was a time where we're spending long hours and sweating over a lot of things and trying to make schedules and all that kind of thing, but it was very enjoyable because we all understood, we had the same goals and had the same appreciation for quality. So with only three people you can do that. Once you get too many, it's hard to have everybody <laughs> in quite such unison, but we did pretty well even at Microsoft and bringing people into that same viewpoint and we hired people according to that kind of viewpoint too. We liked creative people and we liked disciplined people. <laughs>

Brock: Do you think now in some, maybe most contexts, an emphasis more on the creativity than in the discipline?

Austin: Well, there's another approach that is often used. I mean I think both are still used, but there's sort of the creative genius approach where somebody is so smart that they barely need discipline. <laughs> They just create these amazing things and there's disadvantages to working with those people often because they might do one thing and then not produce anything for years and they just cause trouble the rest of the time, but you can get successes from it too.

<laughter>

Austin: I've certainly seen it happen, so there's a lot of ways to be successful.

Brock: Well, if you think about just the span from when you were first encountering programming in Pittsburgh as a high school student to the present, have you reflected on how many places software is? Just the volume of software that's out there in the world from that time to the present or?

Austin: Well, the whole computer industry suffers from this order of magnitude thing that human brains don't understand very well. I mean the factors of increases in memory, the factors of increases in processor speed, the factors of increases in amount of code that's been written, all these are ten to the fifth or sixth or something. These are huge numbers that are beyond visualization to humans and there's very few fields that have had that kind of ratio of explosion in possibilities. So I have thought about it. It seems miraculous, that kind of explosion. It seemed miraculous even in the ‘70s and <laughs> it’s gone on for another 30 years. So, 40 years.

END OF INTERVIEW