Oral History of John Sanguinetti

Interviewed by:
Paul Estrada

Recorded: February 26, 2009
Mountain View, California

CHM Reference number: X5162.2009

© 2009 Computer History Museum

Paul Estrada: [I am Paul Estrada], the Chief Operating Officer at Berkeley Design Automation. Today, we're interviewing John Sanguinetti. John has a long distinguished career in both the computer industry and the electronic design automation industry. Welcome John.

John Sanguinetti: Thank you Paul.

Estrada: Tell about your background and family life before college.

Sanguinetti: I grew up in Maryland, a suburb of Washington DC. I had a pretty standard childhood and teenage years. Probably the distinguishing activity that I had when I was in high school was I played trombone in the high school band and I got pretty interested in music. When I went off to college or was selecting a college, I was really looking for someplace that had a really good college band that I could aspire to play in. I wasn't sure if I was going to go into music or if I was going to go into engineering. And at the time, I thought that maybe the most appealing kind of career would be something like a recording technician or somebody who dealt with both music and electronics even though I didn't really have much of a background or knowledge of electronics at that time. This was the 60s; the middle 60s. So I ended up going to the University of Michigan primarily because of the music program and the college band. It was pretty well renowned as the premier symphonic band in U.S. colleges. So I went there and I went to engineering school. Had no idea of what program I would go into. I just was a general engineering student and played in the Michigan marching band and played in one of the concert bands as well.

Estrada: Did you have any influences or mentors when you grew up? Family members in engineering?

Sanguinetti: No, not in engineering. My father was an engineer with the Navy Department in the Civil Service. And so he always wanted me to be an engineer but he didn't really exert a whole lot of influence on me. Actually the mentor that I had in high school who was the most influential was my trombone teacher. And that's actually fairly common with kids who take music lessons. Their music teacher ends up being a surrogate parent; not a surrogate but an adult figure that can be a role model. And that was certainly the case with me that my teacher was in the United States Marine Band and was just a wonderful guy. So I went off to Michigan. Went into engineering. When I was a freshman, I took the first programming course that they offered and it was a one hour course and I didn't understand a thing.

Estrada: Do you remember what computer system you were working in?

Sanguinetti: It was an IBM 7090. We used the MAD Programming language, the Michigan Algorithm Decoder. The book that we had was written by one of the professors there who had been one of the authors of MAD. I had the book. I didn't understand anything in it. When I got finished with the course, I didn't know if a compiler was hardware or software. All I knew is that you typed up your program on these punch cards and you gave it to somebody at a window. Sometime later you would get a printout back and it wouldn't work. Looking back on it, it's pretty humorous that I really thought this was pretty neat. It was the worst grade I got in college that year and actually throughout my college career, I think that was the worst grade I got throughout my whole career. But when I got home from college that freshman year, I got a job in the government, in the Navy Department. They had a program for summer students. The job that I got was in a group in the Navy Department that was doing shipyard management. There was a guy there; the guy that supervised me, was running a project that was attempting to write a simulation program to manage the shipyards and the schedule of ships that would come in for maintenance. It is not widely known but at least back in those days, the scale of the shipyard management was enormous. It was an organization larger than General Motors. They had eight shipyards; four on the east coast and four on the west coast. And all these ships that had to come in for retrofitting and maintenance and all this. They really didn't have any kind of scientific management for the yards. So this young engineer guy who had come from MIT and got an MBA at Stanford, had enlisted as a lieutenant j.g. to get out of the Vietnam War. He was managing this project and so my boss gave me to him and he gave me little programming projects to do in Fortran; so I had to learn Fortran. Again we were using a 7090 and it was a great experience. I really learned how to program during that summer. A lot of the mysteries and fog of my previous exposure were lifted. And when I got back to college in my sophomore year, I was pretty sure that I wanted to be a programmer. So I started taking programming classes and math classes. Then I discovered that they had a program in the engineering school called Applied Math. And its great advantage was it didn't require you to take any engineering courses other than just a few core ones and it let you take a lot of math and programming courses so that's what I did. By the time I graduated, I had taken all the programming courses that they offered and some of the math courses that were important to a programming career like Linear Algebra and Combinatorial Analysis and Graph Theory. And then I went straight on to graduate school. But I left the engineering college and went to Computer and Communication Sciences, which was a relatively new program at the time. It had been started in '68. This was 1970 when I started graduate school.

Estrada: Where did you attend graduate school?

Sanguinetti: University of Michigan. I stayed at Michigan for 11 years straight.

Estrada: Did you think about when exiting undergraduate entering industry?

Sanguinetti: I did. I actually interviewed at Control Data. But I was pretty certain that I at least needed a Masters degree. I thought I wanted a PhD though I wasn't positive about it but I was pretty sure about it. But the Vietnam War also had an influence on things. When I was applying, or when I was deciding to go either to industry or graduate school, if I was in danger of getting drafted, I wanted to try to get a job in a defense related industry. The computer industry generally qualified for that. But towards the end of my senior year, they held the first draft lottery. I used to say I was one of the winners; I got a very low number or high number. And so, that pretty much made it safe for me to go to graduate school. So I applied to a number of different universities and I got accepted at Berkeley and Michigan. I got a National Science Foundation Fellowship at Michigan so that pretty much sealed the deal. And so I just stayed at Michigan.

Estrada: You got a Masters then eventually--

Sanguinetti: Yeah, I went straight through for Masters and PhD.

Estrada: And your thesis work in a PhD?

 © 2009 Computer History Museum

Sanguinetti: It's funny now. My dissertation was on operating system designing methodology. At the time, my real goal was to be the author of an operating system. I thought that was the pinnacle of the programming profession. During my graduate years, I got pretty enthused about simulation. Writing simulation programs was fun. It was, I thought, really intellectually challenging and interesting. And so what I wanted to do in my dissertation was combine simulation with operating system design. My advisor had done some academic work that was analyzing systems of concurrent communicating processes. And all the examples back then, this was the early 70s; all the examples of concurrent systems were operating systems. The THE [Technische Hogeschool Eindhoven] System from Eindhoven I think it was; Dijkstra's System was very influential. Everybody tried to model that in one way or another. So I did a slight variation of my advisor's work, adding some performance parameters to it and trying to do both some formal analysis and some simulation analysis of it; of models, of operating systems. But the thing that really, really appealed to me was some work by a guy named Nielson. I can't remember his first name now [Norm Nielson]. He was at Stanford I think; who participated in the design of an operating system for an IBM 360 model 67. Yeah, I think by that time it might have been a 370 168. But the goal of it was to write the operating system in a higher level language and I think they used Fortran; and to create it as a top down model and starting the initial design was a simulation model. So they would write the model, simulate it and then elaborate the parts that had been abstracted until eventually the thing got down to a complete implementation and the simulation model was the operating system. I thought this was just really neat. I couldn't imagine a neater kind of a technology. And so that was what I was really trying to make; something that was a methodology that would follow that or that would implement that. The reality was the dissertation was successful in that it got me the PhD. When I finished it I said, "You know, nobody is ever going to look at this again." It just didn't work. I still thought it was a neat methodology but I was convinced that this dissertation would gather dust on a shelf and nobody would ever read it.

Estrada: Was it because it was not practical or esoteric?

Sanguinetti: I thought it was both. Too esoteric; the esoteric parts were too esoteric and really too simple. The practical parts of it - it was just a program. It turns out that there was one other academic who did read my dissertation. I met this lady at a conference a couple years later. She may have been the only one. So when I go out of graduate school, I went off and got a job at DEC as a performance analyst.

Estrada: What year was this?

Sanguinetti: That was 1977. So I got to DEC a month or two months before they formally introduced the VAX. It was a minicomputer that was really a midrange computer; it was a 32 bit minicomputer. So I got there shortly before that was released.

Estrada: How did you make that connection? Did you know someone there?

Sanguinetti: No, it was a letter my advisor wrote to somebody he knew at DEC and that got me an interview and that got me the job.

 © 2009 Computer History Museum

Estrada: How big were they and where were they located?

Sanguinetti: Digital was about- my badge number 64,000 something or other. And they had about 50,000 employees at the time. It was in its heyday. There were facilities all over. They had their own helicopter fleet and the big employment perk at the time was getting a ride on a helicopter during the first week of October when the leaves started to change in New England. Everybody wanted to do that. I actually got one ride that way. But the work that I did was always on the PDP-11. The VAX was really just too new to be doing very much in the way of performance analysis or architectural analysis. Actually, that's probably not true. That work would have been useful but they didn't have any kind of organization to do it. So when I was hired, the performance analysis group was just being formed. So I was one of the first couple people that were there doing that kind of work and first things we did were things like analyzing disk subsystems.

Estrada: Was the objective to improve the hardware?

Sanguinetti: Yes. It was almost always to improve the hardware. One of the really interesting experiences that I had there that actually was a lesson I learned pretty well was that a hardware company has a much easier time changing the hardware than they do changing the software; which is totally counterintuitive. Digital had a fair amount of commercial business with the PDP-11. And somebody had written a COBOL compiler for it. And in order to support COBOL, they had implemented some character oriented instructions in the PDP-11 Instruction Set. They had an additional set of instructions that were an option. And they did this in order to speed up COBOL programs. But what they didn't realize was that the COBOL compiler that they had was what was called a threaded code compiler. It put out code that was simply calls to subroutines. And it was dog slow. And okay, these instructions that they implemented made a difference of maybe 10%. But if they had simply rewritten the COBOL compiler to be reasonable, they would have gotten a 2 or 3X improvement in it; or more than that, probably 10X. It was not a whole lot better than an interpreter. Yet Digital with their hardware mindset, well, we'll make hardware to do this. And so they added all this hardware and it didn't do diddlysquat. At least not if you actually measured it.

So one of the projects that I was assigned was analyzing or predicting the performance gain that somebody would get if they made a smart disk controller. There was a customer in Italy I think that wanted to run UNIX with some database system. This was very early days of UNIX being used outside of academia. Somebody somehow it had gotten to Digital that this customer would like better performance. So again, the hardware orientation, some guy said, "Oh, I can do that. I can make a disk controller that will optimize the head movement." And so you do a whole bunch of seeks and I'll order the seeks so that the head just scans nicely across the platter. And so I was asked to give some kind of analysis of how much performance improvement this was going to get over the standard, just random seek pattern. So I did that. I learned about how you did formal model and it turns out that even at that time there was a fair amount known about how to formally model disc access times with both rotational delay and seek time. You could come up with a closed form solution for random access and for- I think there was a closed form solution for ordered accesses. So I wrote this thing, this report and I sent it up the chain. I got it back, this report, with a handwritten note on it from Gordon Bell saying, "This is great work. We need to not lose stuff like this." That's the first time I had ever had any contact with Gordon Bell. I was happy about that. It was a nice 'atta boy'.

 © 2009 Computer History Museum

One of my friends was working in the Mill a floor below me in the R&D group and he had been working with UNIX for some time at that point. I just happened to run into him one day and I mentioned this stuff to him. He said, “You know, I think that UNIX has a head seek optimization algorithm in it.” So we went down to his office and we dug out the UNIX listing and at the time it was only about that big. Sure enough, there was this head optimization algorithm in it. So we wrote a little experiment and ran the thing and you could-- You’re using disks at that time that you can watch the head moving across the platter and sure enough, gave it a random string and this head nicely scanned across the platter. So I wrote a one page memo back to the project manager saying, “You know, you’re not going to get anything at all out of this hardware controller. This guy’s going to run UNIX and it’s going to do the same thing.” That was probably the most significant thing that I did at DEC was killing a project that would have cost them about a half million dollars for no gain. That was my story from DEC. But the thing that I really learned from that was that hardware companies have a much easier time dealing with hardware than they do with software.

Estrada: Do you think the same thing is true for software companies solving problems with software?

Sanguinetti: Well, no. Software companies don’t have that option. They generally can’t say, “You know, I’d like another instruction in this thing.” Or, “If you could give me a DSP algorithm in hardware, that would improve my life some.” They generally don’t have that option. But the hardware companies do. So it’s very much if all your tools are hammers, all your problems look like nails.

Estrada: How long did you spend at DEC?

Sanguinetti: I was only there for two years. I was there from ’77 to ’79. My wife was in a medical residency in Boston. That’s how we ended up being in Boston. She had gotten an internship at Tuft’s New England Medical Center. So when she finished her internship, she went on to residency back in Detroit where she had gone to medical school. So after the two years at DEC, I went back to the University of Michigan Computing Center until she finished her residency. I was really just on the staff there. I ended up being Assistant Associate Director or something like that. The computing center at Michigan was really quite an innovating place. There were some really smart people. They had written their own operating system for the IBM 360 Model 67. The Model 67 that Michigan had was serial number one and they had participated- actually they had done a great deal of the architecture of the virtual memory system of the 360. So the Michigan Terminal System was widely used on campus. This was the research computing facility for the campus. It was used by six other universities around the world, a couple in Canada, one in England and a couple in the U.S. It was a fairly large operation to support your own operating system that was incompatible with everything else. We had to take all the compilers from OS360 and MVS, when it ultimately became MVS, and put interfaces on them so that they could be run under MTS. That was one of the jobs that I did early on. But eventually, we really had a very elaborate time sharing system. It was not real well known around the academic world but it was really very, very powerful system. It would typically support 200 to 400 users on a dual processor 360 Model 67 and eventually we got the Amdahl 470 serial number two. That was a pretty exciting time. That was what introduced me to the Amdahl Corporation. The 470 was really a very neat machine. We got it while I was a graduate student in 1974. I had a fair amount of contact with Amdahl, or at least a little bit of contact with them. Otherwise, that period was kind of undistinguished. I really didn’t do anything all that significant, I don’t think.

 © 2009 Computer History Museum

Estrada: Your contacts with Amdahl, you eventually joined the company?

Sanguinetti: Yeah. When my wife finished her residency, the deal was we had to go where she went when we went to Boston so the deal was I got to pick the next time. I did some interviewing and I interviewed with a guy named Mac MacDougal at Amdahl. He had been at Control Data in the early seventies and he was one of the first guys to actually write anything about simulating computer systems, full computer systems. He wrote an article in Computing Surveys that was in one of the very early issues and I had seen that. He was doing simulation of both the hardware and the software to some extent but it was mostly hardware. Anyway, he was running a very small group. They called it Performance Architecture at Amdahl. He offered me a job. I was really interested in working with him so I took it.

Estrada: Amdahl at the time was making IBM compatible machines?

Sanguinetti: Yeah. Amdahl started out making a PC. In those days PC stood for plug compatible. So it was an IBM 370 plug compatible computer. It was really quite interesting the challenges that they had because it was completely hardwired machine. There is no microcode in it. IBM would make additions to their instruction set periodically to screw Amdahl. They would add something and would announce it and start delivering it and the customers would say, well, what about you guys? So Amdahl would then have to scramble and make some hardware to add into a 470, which wasn't an easy thing to do. When I got there they had been working on the successor to the 470 called the 580. When I got there it was just the beginning of the beta program for the 580. They had done something to address this problem. They had put in a feature that they called macrocode and this was something that was in between microcode and just plain old software emulation. They put a mode in so that it would trap an instruction and if you put a new instruction into the instruction set, they could set this thing up and basically program the macrocode so that it would trap the instruction, go off into this little software routine, interpret the instruction and return. It was a good deal faster than doing it in just a virtual machine style in the operating system. But it was still a virtual machine. That turned out to have a lot of performance implications that they hadn't really understood.

There were a number of things that they didn't understand about the performance of the machine. They had a guy named Mark Goheen who had worked for Mac who wrote a simulation program of the processor, of the pipeline. He understood the pipeline pretty well. What he didn't understand was the effect of cash misses and the miss ratio. They had done a fair amount of work putting together a benchmark suite. This is back in the days, this is 1982 when I joined them. There weren't standard benchmarks or workloads back then. So they had put together a workload that they thought was pretty representative of a standard commercial kind of workload. They would use that. They would benchmark the 5860 with it and they benchmarked it against a 470 and the goal was that this thing was supposed to be twice as fast as a 470. It turned out when they first turned it on it was like 1.2 times faster than a 470.

So in the six months from the time I got there in July till the end of the year, they sped the machine up to the point where it was about 1.95 I think, it was very close to the original target. No single change had more than a 5% effect. They made just an enormous number of changes and this performance architecture group was right in the middle of it. The hardware guys would say, "What if we do this?" Mark would go run his simulation program and say, "Well, you're going to save .2 cycles per instruction over the course of the workload." It was a really exciting time to be at Amdahl but a very frustrating time too because the company had this real feeling of failure. There was a fear of failure then. The fact that they overcame this real challenge I think made everybody feel pretty good about it but it also made them realize that they really needed to understand these systems a lot more and so that over the course of the two years that I was there, that performance architecture group grew by- I was like the fourth person in it. When I left I think there was about eight; some very good people. Interestingly enough, or at least what I thought was interesting, was that 20 years later when Amdahl was now Fujitsu a number of those people were still there.

Estrada: You were at Amdahl for how long?

Sanguinetti: I was there for just two years. There was a pattern here. Two years. Two years. It recurs. What happened was I got there and this is a very exciting time for the first six months. Then the heat starts to go down a little bit. I started meeting people who were very in tune with the start up culture of the valley. I met with this one guy who had worked closely with the guys in the performance architecture group. He was in some other part of the company. He had taken a sabbatical for some- I'm not sure exactly how that worked. When he came back he said, "I'm going to go join a startup." The startup that he joined was some three guys writing a program that was going to be sold to banks. I thought that was pretty interesting. I started meeting people who were in startups or joining startups and I thought, "I'm in Silicon Valley. This is what you do in Silicon Valley. Why am I in a company that's 4,000 employees?" This is actually worth pointing out. At DEC, there were about 50,000 employees. Amdahl was in order of magnitude smaller. But it still felt like a big company. It was much, much more manageable from a personal standpoint than DEC was. At DEC, there were installations of DEC that I didn't even know about. At Amdahl I knew where all the installations were. It was all in one campus. I knew most of the important people, or at least I could recognize them by sight or had heard their name. It was a more manageable situation. But I understood that that was too big a company.

Again, the trombone comes into it. After I had been living in Sunnyvale for a few months, I found out about a local community band that basically anybody could show up for their rehearsals and play in this community band. It was in Los Altos. So I did that. First day I got there I sat next to a guy, Jim Jaffe, who was a couple years older than me and he said, "If you really want to play in a band, come to the Foothill College Band. They have a night school band that rehearses once a week. It's much better than this." So I did that. Of course, he was there too. Turned out he was one of the founders of Elxsi. Elxsi was a company that was making a super minicomputer as opposed to a mini supercomputer. The distinguishing feature of Elxsi was that it was a multiprocessor computer. They had systems that were up to ten processors; a strange number but that was it. And they had this very fast bus; they called it the gigabus. It was a gigabit per second; had a nice mix of academics and people with industrial experience. The machine had really just been delivered at the time that I got introduced to the company. The company was five years old. It felt like a startup. They thought of themselves as a startup. It had about 350 employees. They offered me a job. I didn't really know if I wanted to be in the software group or the hardware group. I think I was actually hired by the software guys but I did most of my work with the hardware guys. Eventually I ended up being transferred.

I recognized that this was the right direction; 350 was another order of magnitude smaller. It was fun. I was the only performance architect or performance related person there. I ended up doing everything that we did at Amdahl except that I was the only one to do it. I wrote the simulation program that would model the instruction execution unit. I put together a workload to benchmark it. I wrote some display code to show what the real time performance was -- a monitor. It was fun to do. But it was pretty obvious after I'd been there for a while that a five year old startup isn't a startup anymore. This company was a zombie. It wasn't dead but it wasn't living either. Their machine was an ECL machine so it was expensive. It did a few things really well. One of the things it did well was run SPICE. Somebody there [Steve McGrogan] had parallelized SPICE to run on this parallel machine. The machine was actually not a great parallel machine because the processors were too far apart. It used a message passing system to communicate. But you could run parallel programs on it. I actually wrote a parallel sort just for my own education. This was the first parallel program I ever wrote. You could actually get some pretty decent performance improvements. One of the guys there did this with SPICE. It was actually quite good. We had a couple of customers, at least one, that bought a machine to run SPICE on. There were a few other applications that it was good at but by and large it was too expensive for what it did. It just didn't sell that well.

Interestingly enough, if you remember Trilogy, which was a company started by Gene Amdahl and his son [Carlton], they had started a couple years before that and had intended to make a wafer scale machine. Eventually that didn't work. We interviewed one of their architects who told me about their 26 stage pipe. At that time a five stage pipe was standard and a seven stage pipe was pretty deep. When I heard that I said, "Boy. I know why you guys didn't succeed." He agreed with that. They shut down their operations. They had actually gone public. I'm not sure exactly when they went public but they had gone public and raised a whole bunch of money. They had like $40 million. Well, Elxsi had a product, was running out of cash so the first time I actually met Gene Amdahl was when he was there talking to Joe Rizzi the CEO of Elxsi about a merger. Joe had me show Gene our system running a workload that I had put together with this monitor that showed what each of the processors were doing in real time. It was kind of neat. He was sort of politely interested. Ultimately, they did a merger. This was a merger where one company has cash and the other has a product and they call it a merger.

It was pretty apparent though that this wasn't going anywhere. So I got called by a headhunter in March of 1986. The guy said, "I've got these guys who are a client who's looking for a performance analyst or architect or something like that. I got your name. Would you be interested in talking to them?" I said, "Who are they?" He said, "Well, this is a company that is called the Dana Group. It was founded by Alan Michaels and Ben Wegbreit and Gordon Bell." When I heard that I thought, "I saw an article in the paper about these guys." When I saw that, I said, "Boy, they're not going anywhere" because I had a pretty skeptical opinion of this kind of a startup, this kind of machine and the names that I saw in the paper, the ones that I recognized, the only one I really recognized was Gordon. He had just come off on the failure at, I think it was Encore. I don't remember. We'd have to check that. [this is correct]

Estrada: Yeah. Okay, you can go ahead.

Sanguinetti: So, anyway I went and interviewed at the Dana Group and there were 23 people there, actually there were about 20 at the time I interviewed, and I completely turned around my opinion about this. And I think it was simply because they were interested in me so that was flattering and that made me have a good opinion of them. But I interviewed with Gordon Bell so it was you know, my only previous contact with him was having this note scrawled on my report and now I get to meet the guy. And we really had a nice conversation, of course with Gordon conversations are very, very interactive, you have to work to follow his thought pattern and his speech pattern because his mind goes faster than his mouth does. But I just had a great time interviewing with these guys, I met Gordon and Ben Wegbreit

 © 2009 Computer History Museum

and Richard Lowenthal and Allen Michaels, but that was very brief. And they offered me a job so I said, "Okay, I'll do that." Now this was a real startup, so this company had been going for six months and they had a plan, they had a machine on paper, more or less architected and they really didn't have a clear idea of what they wanted me to do. They wanted me to do performance analysis but that really didn't last a whole long time because the machine was almost ready to start implementing and so I did some bus analysis and we–- they had their own bus so I did a simulation program with the bus protocol.

The target for this machine was to do eight megaflops on Linpack and so, you know, in my simulation model I showed that there was some bottleneck in the bus and you wouldn't get eight megaflops on Linpack and so we figured out–- I figured out with the architects how to make some changes to the protocol and sure enough it would get to where it would stream data at its maximum bandwidth. So when I was done with that, that was about two months, they were ready to implement the machine. So this hardware group–- I was part of the hardware group, there were six or seven guys to do all the logic for this graphic super computer, the thing had a processing unit that had a vector unit that had a MIPS processor integer instructions and it had a custom floating point processor that was a vecter unit and it had a graphics subsystem and an I/O subsystem and they had you know, a handful of people to do all this stuff and it was a big job. So we had our first meeting, Richard Lowenthal was the manager, Jon Rubinstein was one of the guys doing the implementation, Glen Miranker was nominally the architect and he was doing implementation too. And everybody said what they were going to do and they looked at me and said, "What are you going to do?" and I said, "Well, I don't know, what do you need?" And they said, "How about design verification, we need that." And I said, "Well, what is that?" And they said, "Well, it's simulation you know, we simulate the netlist and we give it test vectors." And I said, "Well, I know simulation, I can do that." So I signed up to do design verification and that was really a career changing moment. But my first job was to select a language to do the simulation modeling in and…

Estrada: What was the state of the art at that time in terms of design verification. If you went up to other designers and hardware guys said design verification, what would they think?

Sanguinetti: Well, it actually wasn't a commonly used term at the time but the state of the art was that you did your design in schematic capture, so the choices were either Mentor, Daisy or Valid and we used Valid workstations so that the designer would, he would do his hardware design in you know, on a screen with schematics. He would push a button and he would get you know, run a program and he would get a SCALD format for the netlist and basically it's just one-to-one from his drawing. And then he would sit there and construct test vectors. So you basically had to write the test vectors by hand and there were I think, a few macro kind of things that you could use in the user interface that would help you construct test vectors. But basically you had a string of ones and zeros that were going to be applied to all the input pins on a given clock cycle and you had a string of ones and zeros that were expected to be seen on the output pins on that same clock cycle. And that was design verification.

Estrada: So you literally thought through the stages of logic…

Sanguinetti: Right.

Estrada: …and created all the cases that you thought were important.

Sanguinetti: Yeah. And that was hard. And you know, it didn't surprise me when I would find out that the guys that did this didn't do it very well. So when I was given this job the first thing I was going to do was I was going to write a simulation model of the whole machine. You know, I can do that, I thought that was interesting and that was the first time that I harkened back to my dissertation and said, "Maybe there are some ideas there that aren't so bad." And the first job was to pick a language that I could do this in and the choices at the time were Endot or Verilog and Verilog had just been released. The Verilog simulator was just released, version 1.0, and it was competing with Endot. And Endot was a language that was influenced by VHDL a little bit, not a lot though I don't think. But it was a much more event oriented kind of a language than Verilog was. Verilog really looked like process oriented simulation language. I used to say it was kind of a poor process oriented simulation language but it was it was one none-the-less. If you compared Verilog to Simula it was recognizable. Simula is much richer but Verilog was much more oriented towards hardware design and, you know, had a bunch of features that were important. But the thing that appealed to me about Verilog was I looked at it and I said I know I can write a behavioral model of a piece of hardware using this language. And I also know that I can take a netlist and translate it into this language, because Verilog had a netlist syntax and I knew that because we were using these Valid workstations, Valid had this SCALD format for the netlist, I knew that I could write or somebody could write, a translator that would go from SCALD to Verilog. So I picked Verilog and that was really the only reason -- as I looked back on it I never thought that I had great reasons for picking Verilog. I thought that some of it was simply intuition and my own biases. But it turned out that it was the right choice. So the first thing that we did was that we went and hired a contractor to write this SCALD to Verilog translator. And the guy that did that ended up selling the translator back to Gateway–- Gateway was the vendor of Verilog, the creator of it. They ended up contracting back to him maintenance of it and this guy made a living for about 10 years selling this, it was called Vcmp, it was a netlist translator. So that was–- one of my influences on the industry was I got this guy into it.

Estrada: And at that time, so there was a recognizable electronic design automation industry, there was Valid, Mentor…

Sanguinetti: Right. The industry was recognizable. This was my first introduction it, I actually got one of the trade magazines and somebody had written an article about all the EDA companies and had this table of them and interestingly enough one of the reasons I chose Verilog was because I said we're going to need some real, you know, if we're going to simulate this thing we're going to need some real computing power and in that table it had an entry for each of the companies and it listed the platforms that they ran on and for Gateway it said it ran on a Cray. And I thought that sounds good, I don't have a Cray but I can rent time on one because there was a service bureau–- I think they were local, that had a Cray and said they would sell you time on it to run Verilog. It turns out Gateway never actually did that. They had announced it but they never did it. But it turned out that we didn't need it either. We ran on a Sun-3 and it would run about one or two cycles per second, simulated cycles per second or real time and that was not fast but it was fast enough to go through sort of small scale logic…

Estrada: So the EDA industry, the Electronic Design industry, automation industry was then already in the transition going from vendor specific platforms to software that was independent of a platform…

Sanguinetti: Right, Mentor was kind of the lead in that, they sold their own hardware as did Valid and Daisy and Mentor I think were the first to switch to using Apollo's and Sun's. And Daisy, I'm not sure exactly when Daisy went out of business. I think they were always selling hardware and Valid was selling

 © 2009 Computer History Museum

hardware and it took them a long time to get off of that. The Valid workstations that we used at Ardent were PCs. They had a plug in board that you plugged into the PC and that made a Valid workstation or a Valid PC workstation. They had a much more expensive 68000 based machine that I had actually seen used at Elxsi but these PC ones were just as capable. They didn't work quite as well, PCs were pretty slow back then, but it worked okay for schematic capture. But this was the time–- this was 1986, electronic design was still being done you know, just straight schematic capture, there wasn't any higher level design, there wasn't any logic synthesis to speak of and we did our first machine in this style.

So what we would do is I would write a behavioral model of the components. We had the Integer instruction unit which was a MIPS processor. Well we got a model from MIPS —an instruction set accurate model and I wrote a behavioral model of the interface which was the piece that we were designing. You know, we called it Kit for kitchen sink, it was an ASIC that sat between the bus and the processor and it also had the interface to our vector unit. So this was a memory mapped vector unit so you would write to an address and this interface chip would see the address and route the data that was coming along with the address off to the vector unit as an instruction. So then we had two big ASICs in this vector unit along with three Weitech floating point chips that did the floating point add, multiply and divide and we had this big crossbar switch that would sit in between the bus and all the local storage that drove the vector unit. And so I wrote behavior models of all the things that we were designing and put together this model of the processor and we could actually run programs on it. That was pretty slow and so we mostly ran essentially the equivalent of test vectors out of the processor and I wrote a test language with a little interpreter that would take instructions that said something like read address 0x<something or other> and then write this value to some other address and then check for this register to have that or check for this port on the chip to have that value. And that worked pretty well, we could drive data through the whole thing and we could actually run Linpack very slowly, but we ran one loop, one trip through the inter-loop of Linpack in 24 hours. And so that was the–- well we had this behavioral model of the whole thing so when the guys would finish with their hardware design they would give me the netlist and I would take the behavioral model out and plug the netlist in.

And that was where the fun began because that was where we you know, that's where you're doing design verification. And one of my favorite memories was sitting in front of my Sun-3 workstation with Jon Rubinstein on debugging his 780 gate bus arbiter chip of which we had three of these things on a board and the logic was pretty complicated and I had written this behavioral model of it and my behavioral model worked and we plugged his netlist in and it didn't work. And he swore it was right and so we sat there and debugged it together and discovered that we had a mismatch of understanding what the spec meant. And that's a pretty common theme throughout design verification where people didn't understand the spec the same way.

Estrada: Yeah.

Sanguinetti: And this model really was the thing that let you deal with that.

Estrada: And where was synthesis at this point, did…

Sanguinetti: Synthesis was still coming.

Estrada: Yeah.

Sanguinetti: There was this company called Silicon Compilers that Gordon had some contact with and we went over, he and I and one or two other guys, went over to Silicon Compilers' office sometime in '87 or '88 to hear what they were doing, but they didn't do Verilog and we weren't really sure just how usable that was going to be. But in '88 Synopsys came out with Design Compiler and Deirdre Hanford came over to our place with a tape of Design Compiler beta release and handed it to me and she and I stuck it into a machine and got the thing up and I didn't understand really what it was all about, actually I understood it from a conceptual level but I wasn't a guy who did hardware designs so I really wasn't going to be the one to use it. But the other people at the company were very excited about it, they had–- I don't know where they heard about Synopsys but they were convinced from the very beginning that this was the right way to do hardware design and so in 1988 we were just beginning, we had just finished–- this was like April of '88, we had just announced and delivered the first machine. And it was, technically it was a pretty impressive computer, commercially it wasn't successful and it was a real object lesson that so many smart people could work so hard and fail completely. At that time it wasn't apparent that we were failing but it was apparent that we were struggling. So we started to design the next generation of machine and that was coincident with getting this beta release of Design Compiler. So now we had Verilog and we had the Design Compiler and you know, we had the beginnings of what has become the standard design flow.

Estrada: And how did you go then from there until your first venture into EDA?

Sanguinetti: Well, it took a little while, I did a lot more design verification than modeling using Verilog, I really got to be quite an expert at, at least I thought I was and we had a couple of other guys that we had hired, one was Peter Eichenberger he had gotten his PhD out of Stanford and he was doing design verification. He had a really good understanding of hardware and very smart guy. And we did a lot of modeling and we actually, Peter and I shared an office while we were making the second machine and we had always, I had always thought that our interface to our vector unit from the MIPS processor to vector unit had one cycle of latency more than it should have and that was important at least it seemed like it, and so I offered to Jon Rubinstein, who was now the project manager of this machine, I said, "Let Peter and I do this interface logic and we'll use Design Compiler and we'll do it in RTL and we'll synthesize it and we'll get it right." Well, that was a real object lesson because I didn't know anything about RTL, I didn't know anything about what the constraints on Design Compiler were, I thought I'll write the code the way I write my behavioral models and I'll give it to the Design Compiler and it will work. Well, it didn't work that way.

Estrada: RTL is Register Transfer Level…

Sanguinetti: Register Transfer Level coding and it's a style where every clock cycle in the logic is represented, all the activity or all the logic of that clock cycle is represented by some instruction but the clock cycle boundaries are identified. So in Verilog, you say always at positive edge of clock and some block of code. And all of that code is going to get executed when the clock has a positive edge.

Estrada: And the more code that's in there the longer the clock cycle is going to have to be in order to…

 © 2009 Computer History Museum

Sanguinetti: Right, if you have a whole lot of code in there it's got to fit within a clock cycle; if it doesn't fit it doesn't work. And you don't really get much help in figuring out how to fix it. This was actually the biggest problem we had with our first machine and this was, we were still doing schematic capture but our first machine we couldn't close timing on it, we had too much logic in clock cycles. And eventually we solved that problem by a legislative fiat that said that you can't have more than 13 logic levels in a clock cycle. And that actually worked. It was a blunt instrument but that worked. As technology advanced that kind of solution doesn't work to well. So anyway, Peter and I worked on this interface logic and eventually I just gave up and I said, okay Peter go do it. And he had to figure out all the restrictions that the Design Compiler composed on the code and we eventually made his RTL version match my behavioral version and we got the latency to what we wanted. And that was a pretty educational experience. So then, as we go forward, I was pretty obnoxious may not be the right word, but I was pretty opinionated about Verilog-XL. Verilog-XL was the product that came from Gateway. Gateway sold themselves to Cadence in 1989…

Estrada: And this was a simulator that simulated the Verilog program?

Sanguinetti: Right, when we talk about the design language, about Verilog being a design language, back in the '80s it was synonymous with the simulator that implemented the design language. There was only one, Cadence or Gateway had made their documentation trade secret and so it was illegal for anyone else to write a Verilog simulator or analyzer or anything that used Verilog without Gateway's permission. So Gateway had licensed it to Synopsys, so Synopsys could read Verilog and you know, they did understand that the more other products that were surrounding their simulator was good for them, create a real design flow but they weren't really on board with that. They felt that they wanted to keep tight control over Verilog and they wanted to be the only Verilog vendor. And that was a big impetus for VHDL, because you had this budding electronic design automation industry that had a bunch of other companies and only one could sell a simulator, at least a Verilog simulator, so what are the others going to do? Well, they'll go sell a VHDL simulator and try convinced everybody else to use VHDL. And that is what was going on back in those days.

But VHDL was still pretty immature, it wasn't really very usable and there were a handful of VHDL simulators in the late '80s and early '90s but almost none of them had completely implemented the language because it had some really tough features in it to implement. So it was very non-standard. Mentor's VHDL and ViewLogic's VHDL and Valid's VHDL were all different and you couldn't mix, you couldn't change products, so you weren't that much better off than if you just used Verilog. And in the bargain, engineers were really much more attuned to Verilog than they were to VHDL; it was a much simpler language. It looked more like C even though that it really wasn't that much like C. But the expression syntax was and you couldn't say that about VHDL. And by that time I had been at Ardent now for several years and used Verilog for all that time I was pretty much committed to Verilog as a language, I thought that it wasn't great it was a really practical language and I understood it and that always colors your opinions.

But I was really critical of VerilogiXL you know, Verilog-XL had a reputation of being very fast gate level simulation. And that was actually made its impact or its inroads into the ASIC community -- the foundries basically made Verilog-XL a signoff simulator, largely on the strength of its gate level performance. But its behavioral performance was really awful. And that's what I was doing. I was writing these behavioral models and they were dog slow and I would tell anybody that would listen that any idiot could write a simulator faster than this, how could you write something this slow. In retrospect I was pretty obnoxious about it and then I met Phil Moorby, the guy who actually did write Verilog-XL and discovered that he was a really sharp guy but when he wrote logic simulator he wrote it the only way he knew which was as an interpreter. I had back in my experience at Michigan I had learned a great deal about stimulation and all my exposures to simulation was with compiled simulators. And I actually had a very relevant experience; I had a close friend from the computing center who had been a GPSS expert. GPSS was one of the first general purpose simulation languages that was developed by IBM in the late '50s and they had an interpreter for the simulator. And he had the same feeling that this was very slow and he could write a compiler for it and he did and so he had a company named Wolverine Software, they're actually still in business and he wrote a GPSS simulator that was 10 times faster than IBM's GPSS. And I was very well aware of this and this was something that he did in the '70s and I had often talked to him about it and it was you know, we pretty much understood that a compiled simulator was going to run 10 times faster than a interpreted one. So when I found out from Phil that that was how Verilog-XL was implemented, I understood what was going on now and you know, that was nice but there was nothing I could do about it because it was illegal to write a Verilog simulator if you weren't Gateway.

Well Gateway sold themselves to Cadence in 1989 and I think that deal closed in December of '89 and Cadence saw the writing on the wall. This was I think Joe Costello was the one who made this decision, he said, "We've got to open the language up or we're going to get swamped by VHDL" because you had the bulk of the industry focusing on VHDL and only Cadence now trying to promulgate Verilog. So I'm sitting at Ardent this is the change from '89 and '90 you know, Gateway has been sold to Cadence–- I didn't know that that was going to have any affect. Ardent had merged with Stellar, our main competition, and we were still struggling and in fact losing money and we ended up being one of the most, as Steve Blank has often said, one of the larger craters in valley history–- I think we went through about $160 million before the company folded, but as I alluded to earlier it was a real object lesson that you could take a whole lot of very smart people, by far the smartest group of people I've ever worked with, who worked very hard, the average I think was 60 hours a week and we had people who worked 70-80-90 hours a week. A lot of people say, oh I'm in a start up, I work hard, I work long hours. And if you actually add up the hours that you work it turns out that you don't, you're not all that efficient. You don't actually work your 10 hour days or your 12 hour days, maybe out of the 12 hours you work you know, maybe nine or so.

We had people that really worked–- I actually did count up my hours a number of times and I generally worked about 60 hours a week. In fact when we first started to make our machine the guy that had been hired to manage the hardware group was a guy named Bill Worley who had come from HP, he had been at IBM he was actually one of the, he was an architect at IBM for a while, he was really enthused with the startup mentality and startup ethic. He took a piece of paper around to every engineer who worked for him and said, and there was a statement that said, "I promise to work 60 hours a week until this machine is done," and then a place for your signature. And he brought it to me and I said, "Well I can't sign that," you know, "When I was hired here I was very clear that my wife's a physician, she's on call every other night, she's on call every other weekend, I've got a 6-year-old daughter, I can't work, I can't promise to work every weekend and I can't promise to work 60 hours a week." And he said, "Well how much can you promise?" And I said I can promise 55 and so he said, "Okay." He didn't cross off the 55 he just went away and about three days later he came back to me and said, "I've solved your problem." And I said what was that? The problem of your working 60 hours a week, I've got a Sun-3 workstation you can take home, you can do the other five hours at home. I said, well, "Okay."

 © 2009 Computer History Museum

But it turned out that in that first year I told people when I interviewed I can't promise to work Saturday's, my family life just doesn't allow that. And so the first week that I was there Ben Wegbreit came up to me on Friday and said, "Well what have you been doing?" And I told him I'd been writing this simulation program and he said, "Oh, can you show that to me tomorrow?" So I came in that day and I brought my daughter who had just come from soccer practice and so I came into the office and found Ben and showed him my program listing and explained it to him and then I looked around and everybody else was there. And I said, "Geez, this is different," and so I came in the next Saturday and the same thing. It turned out that I worked 44 of the next 52 Saturdays. And that was introduction to startup life. And I found out that I enjoyed it. It was by far the most I had worked in my career, I'm not sure I averaged 40 hours a week before that. But it was a great experience; you had all these smart people you know, Ben Wegbreit, Gordon Bell, Richard Lowenthal, Jon Rubinstein, Randy Allen, Steve Johnson the author of *The Portable C Compiler.* We had graphics guys Rodney Strock, I think he had come from Pixar. We had really smart people who went off to do lots of wonderful things, we worked really hard and we failed completely The lesson that I took away from that was, if you're not doing the right thing it doesn't matter how hard you work and so you probably shouldn't try to kill yourself unless you're sure you're doing the right thing.

Estrada: So then how did you make sure you were doing the right thing?

Sanguinetti: Well, that was the problem, because I didn't know what the right thing was. Here we're in the spring of 1990, the company is cratering and my friends are starting to leave. So Jon Rubinstein who I had gotten really quite close with at that point goes off to NeXT, he gets hired to be the project manager of the NeXT RISC Workstation [NRW] and when he went there, when he went to interview he came back to me and said, "If I need you, will you come?" And I said, "NeXT sounds like an interesting place, but tell me if it's worthwhile doing?" And he said, "Okay, I'll go and I'll let you know." So about two months later he comes back to me and says, "Yeah, I want you here and this is a good place to be." So I went and I interviewed and that was when I first met Steve Jobs. They showed me a demo of the NeXT machine, the NeXT machine had been first delivered I think in '88, The Cube, and they were working on a follow-on workstation that was just a derivative of The Cube, it was a 68040 based machine, the original Cube was a 68030 based machined but they were really working on this RISC Workstation and that was Rubinstein's project. It was going to use a Motorola 88110, 88100 I think was the original processor was going to use, a RISC processor. But they showed me the NeXT machine and NeXTStep and I was simply blown away. It was a computing environment that really looked sophisticated, both professional, polished and it was UNIX and compare this to my Sun workstation and Sun workstation was really quite nice and I had been using that for four years now but this NeXT machine, this was sexy, it was good. And the people there were good. I was obviously impressed with Steve but I knew Rubinstein, he's just a great guy and so I said okay, I don't know what else to do, I did some cursory interviews with Intel and Apple because Mac McDougall had gone to Apple at that time, but I didn't know what I would do at Intel or Apple and I did know what I would do at NeXT. And you know, I was going to do design verification and I was going to use Verilog and I was going to use this sexy new computer. So I did that.

Estrada: I'm going to stop and change the tape.

Sanguinetti: Okay.

 © 2009 Computer History Museum

Estrada: So you're at NeXT Computer now.

Sanguinetti: Yeah, so here I am at NeXT. I'm doing exactly the same thing I was doing at Ardent, doing design verification, writing behavioral models, system models but now we're not plugging in the gate level models, we're plugging in the RTL into the system models but they're still running slow and I'm still complaining. In September of 1990, I saw a little article in the *EE Times*. It was one of these little two paragraph things at the bottom of, I don't know, one of the inside pages, that had a little headline that said, "OVI postpones first meeting." I'm looking at it, what's OVI? And so the first paragraph said, "Open Verilog International, an organization that was started by Cadence to put Verilog into the public domain, has postponed their first meeting." And the second paragraph went on to say that Open Verilog International had been formed by Cadence in May of that year and that was basically the gist of these two paragraphs. I saw that and it was just like that. I said, "This is the idea I've been waiting for."

I had been basically miserable for about six months. When I left Ardent, I had this terrible feeling of being at sea. I knew that NeXT wasn't the right place to be in my progression of companies from DEC to Amdahl to Elxsi, each one was an order of magnitude smaller. Dana was an order of magnitude smaller, it was 24 versus 350. I was number 24. The right number was two or three. Actually, I thought the right number was three for the next one but NeXT was a 400 person company. I knew this wasn't the place I was going to be but I didn't know what I should do and I didn't have an idea for my own startup. I didn't have a good idea for somebody else's startup. I was just at sea. I saw that article and I said, "I know what I can do. I can write a Verilog simulator, a compiled Verilog simulator." It's legal now or it's going to be legal. It actually wasn't legal at the time but the statement, you know, of intent was something I was willing to bank on, that Cadence would, in fact, open the specifications to the language and that's all I cared about. So I started work at home at night to write a prototype. I took a NeXT machine home and I wrote a parser, a top-down parser, for Verilog and I got it almost right. I wrote a prototype to generate the code. First, I generated, you know, I hand compiled a couple of simple Verilog models into some C code that would look like a simulation model, you know, where each subroutine was an event routine. It had a scheduler and a little runtime kernel and got some pretty astounding results. With these simple hand-compiled models, you could run a model that would run 100 times faster than what VerilogXL would do. So I wrote the compiler, you know, prototype compiler to actually do that compilation and, sure enough, I could get the same results I could get by hand. Now, these were simple models and, of course, it's a lot easier to do something, a subset of a language than it is to do the whole thing, but that was encouraging.

It took me about two months to write that prototype. I wrote it from 11:00 p.m. to 1:00 a.m. each night and, with that, I could go and try to recruit people. Two of the guys that I had worked with at Ardent in design verification, Greg Walsh and Peter Eichenberger, who were my two targets, so I first went to Greg and I said, you know, what do you think? Would you like to join me doing this and he said sure, that sounds like a great idea. So this was about November 1990. Then we started looking for the third person and decided that, you know, Peter was a good choice. Peter was actually working for me at NeXT as a consultant. He had gone off to Intergraph. He had been at Intergraph and so he was working as a consultant at NeXT for me and so I saw him every day. I showed him the results of my prototype and I said, you know, I want to do this and he was skeptical. He said, "It's not going to work." So I said, "Well, here, take this home and study it." It had an event trace and it was a number of pages long. That was a Friday. On a Monday, he came back and said, "Okay, you convinced me, I'm in." So we spent the next five months planning to do the company. In that time, I talked to all of my friends and all the people that had influenced me.

The first person I talked to was Gordon Bell and I talked to Alan Michaels, who was the CEO of Ardent. I talked to just about anybody that would listen because I didn't know what I was doing. I had never started a company. I had never managed anything more than about two people. In that time, I talked to Ben Wegbreit, along with Greg and Peter. Ben was appropriately skeptical. He was actually more than appropriately skeptical, he was overly skeptical. After we had that meeting, he went to Greg and said, "I'll give you $20,000 a month to be a consultant here," and so Greg dropped out. So that left just Peter and I. It was a real come to Jesus moment, you know? Peter had been a consultant at NeXT and was now going to stop doing that and he had no income. He had gone off to Switzerland on vacation to spend some time with his father, who lived there. It was right after he left that Greg told me he was dropping out so I went and picked Peter up at the airport when he came back and I said, "Well, bad news, Greg's dropped out." I thought that the chances were, like, 50/50 that Peter would say, "Well, I'm out, too." But he didn't. He said, "Well, okay, we'll just do it ourselves." That was the moment of truth. When we said, okay, we're going to do this. I had told John Rubenstein what I was doing and that I was intending to leave NeXT the beginning of June. We incorporated the company and we started on-- we actually started, I think, on May 27th and I formally left NeXT about two weeks later.

Estrada: This is 1991?

Sanguinetti: This is 1991.

Estrada: Okay.

Sanguinetti: Yeah. And so we got this little office in Los Altos. It had no insulation, a glass wall to the outside. It was springtime so it wasn't terrible but it was, you know, pretty Spartan quarters. My parting move from NeXT, I bought three NeXT workstations along with a printer and Greg had actually done some consulting for a company that had gone broke and they paid him with a Sun-3 workstation and a shoebox disc. He felt so bad about dropping out that he said, "Well, you can take this Sun machine from me." So we put our little network together and I was proudly telling one of my VC friends about it about two weeks later, about how much effort it had taken us to get this little Unix network going with a Sun-3 at the heart of it and these three NeXT workstations talking to it and he said, "How many Ph.D.s did it take to put together a Unix system?" And, well, in this case, two <laughter> and it took us awhile but it worked. We started, for real, writing what became VCS.

Estrada: And VCS stands for?

Sanguinetti: It stands for Verilog Compiled Simulator. The intent was we were going to write a compiler. It will produce C code. C code will be compiled by your C compiler and that will produce an object and the object program, you run that and that's your simulation model. It was kind of an unfamiliar-- it was definitely an unfamiliar paradigm for a lot of people in the EDA industry. Again, I never stopped talking to people. People, nowadays, often, you know, start up sort of in stealth mode and people think that it's very important to keep what you're doing secret. I had no concept of that. I was so unsure of myself that I wanted to talk to anybody that might have an opinion or might be able to give me some good advice. I talked to Ron Collette, who was the DataQuest analyst and he said, "Well, that's very interesting but why don't you do it for VHDL because VHDL is going to overtake Verilog and Verilog's going to die." I said, "Well, I know Verilog and I don't know VHDL." I said, "I know how I can do this with Verilog. I don't know how to do it with VHDL." And he said, "Well, you know, this may be an interesting little project but it's not going anywhere." I talked to Andy Rappaport who was-- I'm not sure exactly what his position was then and he's been in the VC business now for awhile and he had the same comment. He said, "Do VHDL." He said, "Verilog's not interesting. It's going to be dead in a year." And I went and talked to one of my friends, one of the guys who had been at Ardent who was now a partner at Sequoia, VC firm, and I went and talked to him. I told him what I was going to do. I said, "You know, I'm not looking for VC money. I don't think any VC would fund this and I don't think I need it. I can support myself and Peter can live off his savings for awhile. So I just wanted to get your advice." And he said, "Well, you're right, no VC would ever touch this." He said, "We just funded Redwood Design Automation." This was a company that was started by...

Estrada: Doug Fairbairn?

Sanguinetti: Doug Fairbairn, yeah. Doug Fairbairn started Redwood. By coincidence, I had gotten introduced to the guy who was the V.P. of engineering of Redwood and I had an appointment with him following my meeting with the guy at Sequoia. Well, the guy at Sequoia said, "Well, we just funded Redwood. They're going to do $100 million in three years in revenue." I said, "Really? What are they doing?" And he told me that they were doing this combination of simulator and synthesis tool and they were using their own language. They were inventing their own language. So I went, you know, I left Sequoia and drove down to San Jose to meet this guy, who was their V.P. of engineering, and, during the course of that conversation, I said, "Did you really tell the VCs that you were going to do $100 million in three years?" He says, "Yeah," he said, "We really think we're going to do about 50 in three years but we told them 100." And I thought, that doesn't sound like a good idea to me. Ever since then, I was really skeptical about Redwood and I loved competing against them, which we did quite a bit and we beat them every time but there were a variety of reasons for that.

But, anyway, I continued to hear from anybody who would tell me that VHDL was going to overtake Verilog, why waste your time on Verilog? What I knew, as a user, was that, if I was an engineer who used Verilog, I wasn't going to switch to VHDL because there was nothing in it for me to do that. The only thing that I would get from it would be political. I might have other products available to me, you know, other than Cadence but VHDL was slower, in general, even slower than VerilogXL, and it gave you no more power. It really had no more expressibility or expressive power than Verilog did. So I was willing to place my bet on doing a Verilog compiler. Peter and I worked on this for about nine months, a little longer than that. It was about ten months before we got a beta version of it and we shipped the beta version to NeXT, not surprising. You've got your friends there. It sort of worked. It really was much faster than VerilogXL and, for NeXT designs it typically ran about 20 times faster than VerilogXL did. Unfortunately, of course, it had a lot of bugs and we had not implemented a number of features, two features, primarily, tri-state components and specify blocks in the language. Those are things that you didn't normally use in RTL regressions and that was really where we were going to aim this at. We were going to aim it at people who were like us, like Peter and I. We did our design verification, we ran the RTL, we never ran gate-level models. We never did timing simulation. We were making this product for ourselves. Because we did that, because that was what we were trying to do, we could make engineering judgments about things to leave out that, if you didn't have that context, you wouldn't have been able to do.

So there were several other companies, one in particular, that tried to make "Verilog clones" and they were trying to make products that were just exactly like VerilogXL but they were going to sell it cheaper.

They were approaching it from an EDA background. This company was called CAD Artisans. Their product was called AuSim, A-U-S-I-M for gold simulator, and they struggled with it because they didn't understand that the language is not completely specified and that you can write a model that is indeterminate, that, depending on the order of events, you can get different results. Those are generally race conditions in hardware and we took the attitude that, well, if you do that, your model's at fault. You know, the simulator is free to do whatever event ordering that's legal and, if it does it differently from VerilogXL, well, that's okay. They didn't do that. They tried to say, "We're going to make it match VerilogXL" and that's very hard when you don't have the reference simulator to even compare it to or, if you do, you know, you’ve got to look inside to see how it does things or you have to spend a lot of time doing experimentation. These guys, they never succeeded and there were several other companies that were taking the same kind of approach.

We just insisted that, you’ve got to write your model with correct semantics and we'll implement what you should implement. We're not going to recreate VerilogXL's bugs and that was a major contributor to our success, that we made good engineering judgments about how the thing ought to work and then, of course, we also made it fast. It was, indeed, between 10 and 20 times faster than VerilogXL for almost everything it ran except for gate level models. And, for gate level models, it was actually slower.

Estrada: So how long was it from the time you started the company then to when you had a product you were selling and how did that ramp up?

Sanguinetti: It was 12 months from the time that we started, it was actually 11 months from the time that we started 'til the time we delivered our first beta. We actually got paid for our beta. I think we got $25,000 for 20 licenses from NeXT. No, that's not right. It was $50,000 for 20 licenses. So we got a $25,000 check in June of 1992, our first income. Then we had beta at Sun and we didn't-- I don't remember if we got paid for that or not. I think maybe we did but it wasn't a lot. By July-- yeah, that's right, we did get paid because, July 1st of 1992, we had now been working 13 months and we hadn't been taking any salaries. At that point, we had four people, Joel Paston had joined us as an AE and he did a lot of the work on the PLI and we had a sales guy. All those people worked for nothing for some significant period of time. Peter and I for the full 13 months, Joel for about three months of it and the sales guy for about six months of it, I think. We had had this interaction with Sun and they gave us a model that was-- they called it the tsunami model. It was a model with the first microsparc and it was a big model. It was about 100,000 lines of code and it wouldn't work. VerilogXL, you know, ran it just fine. VCS didn't. And so we spent a full month getting that thing to work, ironing out the bugs. When that model ran, we knew that we actually had a product. Up until that point, you're still a little bit nervous that, you know, is this thing really going to work?

As I was mentioning before, I talked to a lot of people in the industry, people that I had gotten introduced to, and it was surprising that a lot of them said, "You can't write a compiler for Verilog. It's not possible." And, of course, I knew that wasn't true. You can write a compiler for any language, it just may be difficult or very crufty but, you know, you can do it. But there was a lot of skepticism that this was even a feasible project and so, you know, until you can actually prove that it's working, you know, you've got some of that uncertainty as well. So, when we got that tsunami model to work, that was when we knew we had a product. That was the middle of July of '92. Sun paid us a little bit of money, something like $50,000 or something, but then, a month later, they said, "Okay, now we want to negotiate, you know, a real purchase." They bought, I don't know, 20 or 30 licenses, something like that, for about $480,000,

 © 2009 Computer History Museum

something like that but it was in three stages. They were going to give us $160,000 up front and then we had this long list of stuff that we had to do, mostly PLI routines because we hadn't done much of the PLI. In fact, with the PLI, we weren't even sure how we were going to do some of them. PLI was, in those days, divided into two sections called the TF routines and the ACC routines. We knew how to do the TF routines. We didn't really know how to do the ACC routines. We knew we were going to have to do some major surgery to get them to work and we really didn't want to do that. I didn't want to do it. But, anyway, we had this long list and they said, "Okay, you deliver all the TF routines, we'll pay you another $160,000. You better do that by the end of the year. We'll give you another $160,000 when you deliver all of these ACC routines." And they had a long list of requirements. So I swallowed real hard and I signed the contract because I really wasn't sure we were going to be able to do that third piece. But, by the end of the year, we had done the second piece and we got paid for it. We then got contacted by Sun and they said, "We would like to do another purchase. We'd like to renegotiate our contract with you." So we said, "Okay" and they wanted 100 licenses and they wanted them through, I think, July of that year and then they wanted to negotiate a site license or something like that. So the first thing I did was I said, "Fine with me. Just let me out of that last commitment," and they did.

But, anyway, we didn't pay any salaries until July of 1992 so we'd worked for 13 months without pay, actually, 14 months. At that point, when we could start paying, we could start hiring some other people and so we hired Mike McNamara and Todd Massey and a couple of other people. Now it's starting to look like a real company. We were up to about seven people by the end of 1992. We did a formal product announcement in November of '92 at the Tech Museum in San Jose and that was really fun. We sent out invitations to the press and to some of our local prospects and we had a couple of customers stand up and tell, you know, everybody how well this worked for them. We had Shrenik Mehta from Sun and Pete Foley from SuperMac and we had Gordon Bell, you know, give a little bit of a talk. Of course, because Gordon was there, that drew people. Gordon was on our board. Our board consisted of myself, Peter and Gordon. We never raised any money. Gordon put in, I think, $5,000 just to buy some stock and that was it. We were off and running. So, in 1992, we ended '92 with five customers and we had done I think a million dollars in '92 but we hadn't gotten paid. We'd gotten POs but we hadn't actually gotten paid for the last two, which came in December. The first week of January, I wrote a check for one of our utility bills and the balance in the checkbook was less than $1,000 after writing that check. About three days later, we got a check for $400,000 from Cyrix and that was when we were off and running.

Estrada: So how did your growth then go in terms of people and revenue?

Sanguinetti: Well, we were at seven people at the end of '92 and we saw that, you know, we were really a business now. We could start growing, hiring people. We still had work to do on VCS but we started advertising in January of '93. I used Steve Blank as my "marketing advisor". I had an advisory board of sorts and Steve was certainly a key part of that. We spent a fair amount of time trying to figure out how we were going to get publicity because *EE Times* wouldn't write about us. Richard Goering, you know, he was influenced by all the opinion makers, who said Verilog is dead, VHDL is ascendant and why write about this little company? Well, he did come to our product launch and we actually did get a little bit of an article about that but we couldn't get any coverage in *EE Times* or *EDN* or any of the trade rags then so we pretty much said, well, we got this money now, let's buy advertising. If they won't write about us, we'll write about us. My recollection is we were really the first simulator company to actually advertise. We were certainly the first Verilog simulator company. Gateway ran a series of ads back in the '80s before they were acquired by Cadence that were very good and I remember they had an affect on me when I was at Ardent and saw these things in *EE Times*. So we thought, well, we'll do that, too.

We ran a series of ads over the course of the next year that ended up being both fun and very effective. We finally got our message right by the third ad. The first ad was really just a notice to the industry. The headline on the ad was, "If the Cadence is too slow, pick up the pace." Our ad guy thought that was very clever and so did we. It was really just sort of a poke in the eye at Cadence. It wasn't a great ad. The next ad, I've forgotten what the headline was. The third ad we got it right. The headline was, "Finally, fast Verilog." And all the rest of the text was meaningless. It didn't matter. The headline was we're fast, we're Verilog. That was our message and we stuck to that message for basically the life of the company. We hired Simon Davidmann in England to be our European representative. Actually, he was an employee. He opened a branch office in the U.K. He managed distribution in Europe and we had quite a number of European customers over the course of that year. We contracted with a distributor in Japan that was, Kanematsu. They were kind of a fledgling EDA distributor. We couldn't really get any interest from the mainline EDA Japanese distributors. We got a distributor in Taiwan that was-- I forget the name that they used at the time [The EDA Team]. The principals of that distributor became the founders of SpringSoft, you know, the company that acquired Novas and sells Debussy. They actually never sold a single copy of VCS but we had that contact with them. The question was?

Estrada: The growth of the company.

Sanguinetti: How it grew.

Estrada: Yes.

Sanguinetti: We grew up to about 24 people in the space of 1993. The most significant addition was Allen Michaels. Toward the middle of '93, we went to DAC. Our salesman quit. We had a really successful DAC and it was really fun. It was very energizing to, you know, we were 13 people in June of '93 and...

Estrada: DAC is the Design Automation...

Sanguinetti: Design Automation Conference where everybody in the industry, you know, shows up and has a booth on the exhibit floor. So we had a little booth and we did the usual kinds of rounds of talking to analysts and potential customers. We came away from DAC thinking, you know, we really have a company. We've got good people, all of our hires were really topnotch people. We knew what we were doing. We were pretty certain we were going to be successful if we did what we were trying to do. But the company had some real holes. I was the only marketing person we had. Steve was my advisor but, you know, I did all the ad copy. I wrote whatever had to get written for the trade magazines. We were just starting to build a sales force. This is where we discovered the value of a pipeline, which we didn't have and, of course, our sales were really good for January and February and then fell off a cliff for the next couple of months because we had exhausted all of our early contacts and didn't have any salespeople to build them up yet. Like I said, our sales guy quit though he had hired his replacement before that. So I contacted Allen Michaels, who had retired and was living in Phoenix or Tempe or some place in Arizona and asked him if he would be on my board of directors because I really felt like I needed general business expertise. So he said, "Well, I'll come up and I'll talk to you and see what I think." So he came up to the Valley and spent a day with us. He said, "Well, you need a V.P. of sales." I said, "Well, I know that, I got a guy that we had worked with at Stardent lined up for an interview for tomorrow.

He's going to come in." So Alan said, "Okay, I'll stick around. I'll interview the guy." So the guy comes in and he interviews and he leaves and I'm ready to hire the guy. He's a great guy. Alan says, "Don't hire him. Hire me." I almost dropped my teeth and I said, "Let me think about that." So, next day, I said, "Okay, you're on. What title do you want?" He says, "Oh, I don't care. Call me anything." He says, "Call me Chief of Operations." So that was the title we gave him, or V.P. of operations, I think. So he put together the sales force and he managed the sales force. We had a sales force of just three people. It was three local sales guys plus Simon plus the Japanese distributor. That was what we ended '93 with.

So, by '93, we were up to about 20 people, by the end of '93 and we had done five and a half million dollars in sales. We had done a site license with Sun. That was really what put us on the map. One of the lessons that I learned from this was that, if you could get a high profile, credible customer that gives you a lot of credibility in the market. In the EDA world, credibility is everything. If people believe what you say, they'll cut you a lot of slack. But, if they don't believe what you say, they won't give you a chance. Of, you know, my attributes, I generally don't talk all that much, this interview notwithstanding, but, when I'm out talking to customers or to prospects, I don't talk a lot but, generally, at least in those days, I definitely knew what I was talking about and I had credibility and that was my most important attribute. I wasn't all that great a CEO. I didn't even have that title. My title was president. I really enjoyed doing some of the stuff and I was good at some things but by far the most important attribute I had was that people believed me and for good reason, or at least I thought so. I had written half of VCS. I knew what was there.

Estrada: Yes.

Sanguinetti: In 1994, Allen Michaels had been with us for several months and he didn't know anything about the EDA business before he started with us. So, once he joined, he started asking people he knew and asking around about the industry. He came up to me one day in the fall of '93 and said, "You know, EDA is a shitty industry." He said, "Companies have their day in the sun and then they fade." He said, "We've got our day in the sun. We better make hay while the sun shines." That's almost an exact quote. And so he said, "I know an investment banker. Let's go public." So here we are with five and a half million dollars of revenue for 1993 and we're thinking about going public. So we talked to Unterberg Harris, because they were buddies of Allen's, and I think the guy's name was Carl Unterberg, he was one of the principals and he came and we told him the story. He said, "Yeah, I can take you public." So we started that process in January of '94. We started making plans to go public. It wasn't completely outrageous an idea. There's a company called ISS that had made a design rule checker that had actually gone public the year before on $7 million of revenue so this was maybe a little bit of a stretch but, you know, we were advised that maybe you'll get a $30 million market cap, you know? Maybe 25 but somewhere in that range. So I wrote a prospectus, we hired accountants, we hired lawyers. We had our kickoff meeting and Allen turns to me and says, "Remember this meeting well because you're paying every one of these guys by the hour." <laughter> It was kind of exciting but it seemed like a little bit of a stretch but, okay, you know, we were on this path.

And then we got an offer from Synopsis that said, "Would you be interested in being acquired?" And when I had started the company, I was a regular bike rider and I rode with a friend of mine every weekend. He was a VC, sort of a third tier VC, and he asked me, when I was telling him my idea about I was going to do this, he said, "Well, what's your exit strategy?" I said, "Well, we'll get acquired." He said, "Well, by who?" I said, "We'll get acquired by Synopsis. They obviously need the product to compete against Cadence." Well, okay, here it is a couple years later and we get this offer from Synopsis and it turns out they actually did make us an offer of about, I think they offered us, like, $10 million or something like that and we said, "Thanks, but no thanks," and they went away. Well, a little while later, about a month later, we got an offer from ViewLogic and they made us an offer of $18 million. We said, "Well, we don't have any trouble turning that down, either." A couple weeks later, we got an offer from ViewLogic that raised it to $25 million. At that point, our investment banker was telling us our market cap, if we went public, would be between 25 and 30. Allen was saying, "You don't want to be the CEO of a public company, too much work," you know, "You're not the right kind of person for that. We really ought to take the offer. We ought to be acquired." So we started negotiating with ViewLogic. We got this offer on Friday of $25 million and it said, "Sign this by Monday of next week." Being somewhat naïve, you know, I kind of thought that meant something and so I think I called Aart de Geus and said, "I got this offer from ViewLogic. Would you like to talk to us?" And so he came over to my house the next day and we had a really nice conversation. He said, "Well, let me put together a meeting tomorrow." So, Sunday morning, we went over to Synopsis and there were 11 people on the Synopsis side of the table and Peter, myself and Allen on our side of the table. This was, again, one of those things that you don't ever forget. We talked about the virtues of being acquired by Synopsis and Synopsis told us that, you know, if we acquire you, you become Synopsis. You'll move into our building, Chronologic goes away, you know, everybody will get an office. We're not going to fire anybody but the sales guys will become sales guys for Synopsis and you will be Synopsis employees.

Well, ViewLogic had been telling us something very different. They had been saying our view of acquisitions is we'll operate you as an independent subsidiary and all of your people will still be, you know, Chronologic people, even though it'll be under the ViewLogic umbrella. Well, that sounded really attractive because we liked our company. We liked the people we were working with. We were now up to 24 people. We felt like we were being successful and we had momentum and we didn't want that to change. So here we're in this Synopsis meeting and they're telling us that, we acquire you, you're going to get swallowed. ViewLogic is telling us that, you know, we acquire you and you'll still be the same as you are. So Harvey Jones said, "What'll it take for you to say yes?" And I said, "If you offer us $30 million here, on the spot, I'll sign it on the spot." I said, "Anything less, we'll go back to our office and we'll make a decision." And so, during the course of this meeting-- well, I'll get back to that. So, at that point, we go off to another room. They brought some pizzas in for us. They have their discussion. They bring us back into the meeting, so, "$25 million cash." And we said, "Okay, thank you, we'll go back and talk about it." And we went back to the office and said, "Well, you know, same price but we like the ViewLogic model better than the Synopsis model so we'll go with ViewLogic." Well, it turned out that was a big mistake and it was largely-- there were a lot of reasons why we made that mistake and I have to say that it really was my mistake, though I blamed Allen for it a lot because he pushed ViewLogic. It was my naïveté about the EDA industry that made me pick ViewLogic over Synopsis. I didn't know the way mergers worked or acquisitions. I didn't realize that the ViewLogic model wasn't one that was going to be sustained. As soon as the company got into trouble, the natural reaction of management was to try to get more control on all pieces of the company which meant that these independent subsidiaries ceased to be independent. Essentially, Synopsis was simply being honest with us about what was going to happen.

The other issue was that, you know, Synopsis was clearly a well-run, ethical company and ViewLogic really wasn't. They were run by Alan Hanover and Gene Robinson. The sales philosophy, actually, the whole management philosophy was summed up by Gene's statement that he made at one of the sales meetings that said, "Our goal is to stuff our products into every orifice of each of our customers." That's a direct quote. That's the way they ran the company. And, of course, the company essentially ran into the ground. We ended up being very disillusioned by the whole thing and ended up in a pretty nasty lawsuit

 © 2009 Computer History Museum

trying to undo the merger. The lawsuit was a mistake. We shouldn't have tried to do it. We got some advice from a lawyer, fairly high priced lawyer who thought it would work. He thought you could do it. In the end, it didn't work and we ended up pretty much all of the principals of Chronologic left during the lawsuit, that was 1995, and the lawsuit was resolved in the beginning of '96 in ViewLogic's favor so we were basically out of the company and off trying to put together our lives for the rest of the career. One of the biggest regrets that I have from that experience was, in that meeting with Synopsis, Harvey Jones, who was then the chairman and I think he was also the CEO of Synopsis at that time, we were talking about Verilog and VHDL and I said to him, you know, "Synopsis has been the biggest proponent of VHDL in the industry. Here, we're a Verilog company and we hate VHDL. I would just as soon kill VHDL." I said, "What's going to happen if you acquire us? Are you really going to promote Verilog over your VHDL simulator?" And he slammed his hand on the table and he said, "If you join us, we will kill VHDL." My biggest regret-- not my biggest but one of my larger regrets from that was that I didn't take him up on it because it would have done the industry so much good to kill VHDL at that point. Joe Costello, several years later, called VHDL the $400 million mistake and, by now, I think you can say it's about a $2 billion mistake. It was an opportunity lost because Synopsis had the ability to do that. With Cadence pushing Verilog and Synopsis pushing Verilog exclusively, that would have been enough to kill VHDL or at least I think it would have.

Estrada: VCS somehow then...

Estrada: So VCS eventually found its way into Synopsis?

Sanguinetti: Yeah. So what happened was ViewLogic, you know, fell on hard times. They were a public company. Their market cap dropped to fairly low numbers, in the hundred million range, 150 million, 120 million, I think, and eventually, in 1998, I think it was, Synopsis acquired ViewLogic and they did exactly what they should have done. They acquired ViewLogic for three products. It was VCS, Quad's motive and Sunrise test-- I forgot what the product was from Sunrise but those three were ASIC oriented products and all the rest of ViewLogic's products were FPGA oriented. And so what Synopsis did was they kept those three, which were also West Coast subsidiaries and they spun out or basically threw away the FPGA part so the original ViewLogic became ViewLogic again but as a private company. That was the Massachusetts part. The ASIC parts were absorbed into Synopsis. They killed Motive but transitioned all their customers to PrimeTime and Sunrise, I think that product lived for some period of time and I'm not sure what it's morphed into. But VCS was really the crown jewel there and they, to Synopsis' credit, put a lot of investment into maintaining it and developing it, mostly adding features to it. They added Vera to it, they added something called Radiant to it and made it really the industry standard. It is still the leading Verilog simulator in the industry.

Estrada: And Cadence eventually created their own compiled simulator?

Sanguinetti: Cadence, in the early times with VCS, we embarrassed Cadence quite a bit. High profile customers would say, "Well, this thing runs 10 times faster than VerilogXL" and Cadence's response was, "Well, we'll speed up VerilogXL." And so they came out with VerilogXL Turbo and then with Turbo II [TwinTurbo], I think it was. That was really the best they could do and they ended up speeding up VerilogXL by a factor of about two or three. Well, that didn't really make them competitive with VCS so they acquired a company out of Cleveland, I think it was, that had a VHDL compiler and they took that project, took that company, it was a two-person company, and they turned that into what they called NCSim, Native Compiled Simulator, that was both Verilog and VHDL. It took them quite a while to get that to production quality but eventually they did and they transitioned all their customers from VerilogXL to NCSim and that's now Cadence's standard product. It's quite competitive with VCS. There's a lot of models that, if you run both of them, they'll run within, you know, a few percent of each other. I don't know what the overall speed is. I think VCS is still somewhat faster over a larger number of models but it can't be all that significant because, basically, they use the same technology.

Estrada: Then, in the meantime, you went off and started your next venture?

Sanguinetti: Yeah. Because I was out of ViewLogic, one of the big mistakes I made during the acquisition was I signed a non-compete agreement for four years. So I couldn't do anything that was competitive with ViewLogic for four years and ViewLogic felt like they were a mainline EDA supplier so everything was competitive with them. Well, I went off to-- first, all of the people that were refugees from Chronologic started a consulting company and, out of that, three of them went off to start Surefire, which got acquired by Verisity and is now part of Cadence, and the rest of us kind of dispersed. I wrote a Verilog online training course that is actually still on the web. Then I joined up with another guy from Ardent that I knew to do a company called ModelLogic where-- this was his idea. He was a mechanical engineer and he said, "The electronic world has a language like Verilog that you can describe behavior over time. The mechanical world doesn't have anything like that. Why don't we make something like that?" So we tried to make a language that would describe geometry changing over time. It was a neat project. There ended up being three of us. We concluded, after about two years, that this was a nice research project but it wasn't a business so we shut it down at the end of '97.

At that point, my non-compete expired, it was going to expire in April of '98, and so I started looking around for something to do. I headed back to EDA. I had met up with a friend of mind from graduate school days who was living in the valley who told me that he was the chief architect of a company called IX Micro, a local graphics company, and he said, "Well, you know, we do our design in Verilog but we can't really afford to buy Verilog licenses. You guys charge too much." So I wrote a C++ class library that allowed me to write Verilog but write it in C++. I said, "Well, that's pretty interesting. Let me see it." So he gave me a copy of it or gave me the source and I went to Jon Rubenstein, who was now the V.P. of engineering at Apple, and he had just come to Apple, and his first act was laying off about several hundred engineers. I told him about this idea and he said, "Well, you know, I'll give you a contractor's badge. You can come over here and play around with it if you want. You can have access to our models." So I spent the next several months experimenting with this C++ class library and these Verilog models and trying to see if I could take a Verilog model and write it in this C++ class library and see what kind of performance you could get. It was enough so that I believed that this was viable. I never thought Verilog was going to have a really long life, far longer than I ever thought, and I thought that moving to a general purpose language like C++ was really something that was on the right side of the march of technology. So, in the spring of '98, Andy Goodrich, the guy that had given me the C++ class library, myself, and Randy Allen, who had been also at Ardent and had been the author of the Ardent C Compiler, or at least a good part of it, we got together and we started what was called a C2 Design Automation, which was named CynApps or we did business as CynApps, which has become Forte Design Systems. So we started the company in May of '98 to do a high level design environment using C++ and to do a synthesis product from that high level design representation to RTL, which would then go into the standard design flow. So that was the genesis of the company that I'm currently in, Forte Design Systems.

 © 2009 Computer History Museum

Estrada: And the state of that company?

Sanguinetti: We're now ten years into it and we no longer use the original C++ class library. We now use SystemC, which is conceptually just exactly the same thing, just different syntax. We have a Synthesis product that we sell. It's getting increasing acceptance, particularly in the consumer electronics industry, but it's clearly the way-- high level design is clearly the way hardware design is going to go. The open questions are what form is that going to take? This is a form that I think we've demonstrated is viable. The Synthesis product has now been on the market for six years and it works. It works very well. We have many shipping products on the market. You can go to Best Buy and buy HDTVs and DVD players and cameras that all have chips in them that have been designed using SystemC and our synthesis product. Our synthesis product is named Cynthesizer and we have high hopes for it. Ten years of history in the company, there have been lots of twists and turns but the vision of the company has been consistent and we're doing what we set out to do, even though we also set out to do a whole lot more. We've now narrowed it down to just this one product. It works. I'm very hopeful that this is the direction that the industry is going to go.

Estrada: What advice would you have for the next generation of technologists in your wake?

Sanguinetti: The first piece of advice is don't take too much money, don't take too much investment money. If you can-- the great thing about EDA in the early days, back when we did Chronologic, was you actually could take three guys, 18 months and do a product that would work or that had value in the market. At that scale, you can actually fund the development yourself. It's much harder to do that now because the bar is really higher. It takes more than three guys more than 18 months to do a product from scratch now but it still remains that, if you can minimize the amount of investment money you take, you increase your flexibility a great deal. That's one of the lessons I've learned from Forte, you know, because Chronologic, we took no money. Forte, we took a lot and it really does change your approach.

I think the other thing that is really one of the most valuable lessons is focus. The ad that I mentioned from Chronologic, “Finally Fast Verilog”, perfectly captured the focus of the company. It was fast and it was Verilog. It had to be compatible, it had to be legal Verilog but it had to be fast. It had to be both of those things and that's all it had to do. We could blow off everything else. We didn't do specify blocks, which is what is required for timing verification, we didn't do those for two years. We did have to do the tri-states, which we had left out originally. You had to do them pretty quickly but if you focus on one thing, you've got a much better chance of being successful. We learned that the hard way at Forte. At Forte, we started out saying, you know, we need this whole design environment, you know, if we're going to sell a Synthesis product, there's got to be something to synthesize, which means people have to write these models in this higher level language and that's SystemC. Well, we tried to promote our own class library. Synopsis was pushing SystemC. We should have simply said, "Okay, we'll let you guys do that work and we'll just take SystemC." We didn't learn that for a couple of years. We switched to SystemC about 2001, I think it was, 2002. Then we jettisoned everything else and we said, "Okay, we're only going to do one thing." We're just going to do this Synthesis product. It was more than enough to keep us busy. So that I think is a really important bit of advice. You focus and you've got a much better chance of being successful.

 © 2009 Computer History Museum

I think another piece of advice, which I'm not sure how valuable it is for somebody trying to do a startup is something that I heard from Richard Newton in one of the last DAC speeches he made. That was that we don't take enough risks in the industry. We don't try to push the envelope far enough. I think that's generally true. Now, I don't know just how much opportunity you have, as a small startup. If you're well funded, that's one thing. I think Magna is a good example of really pushing the envelope with a well funded startup and they were quite successful. They changed the way people did things. It's hard to think of too many other examples. You know, Chronologic certainly changed the industry. Now, nobody does interpreted logic simulators. They're all compiled simulators, because it's obvious what the advantages are. But it's hard to think of too many other small startups that have really changed the industry with a radically different technology and you could argue that Chronologic wasn't even that radically different. It was still Verilog. So though I agree with Richard Newton's sentiment, that we don't take enough risks, I'm not sure that there's that much opportunity to do that in a startup environment. You know, you're taking a lot of risks as it is.

Estrada: Well, thank you very much for participating.

Sanguinetti: It was my pleasure.

END OF INTERVIEW