Oral History of Lee Lorenzen

Interviewed by:
Dag Spicer

Recorded May 24, 2017
Mountain View, CA

CHM Reference number: X8217.2017

© 2017 Computer History Museum

Table of Contents

Lee Lorenzen

Conducted by Software Industry Special Interest Group

Abstract: Lee Lorenzen first reviews his early childhood years, moving frequently with his mother, siblings, and father, who was a career Air Force Colonel. After spending a few years at the Air Force Academy, Lorenzen transferred to Southern Methodist University and graduated with a degree in computer science. His early career included positions at Xerox, where he developed the Lone Star graphical user interface, and Digital Research, where he developed the GEM (Graphic Environment Manager) operating system. Lorenzen's involvement in desktop publishing included his first startup, Ventura. Following the sale of Ventura to Xerox, he founded and actively invested in a number of other startup computer companies, including Altura Software and the more recent Catalog City.

Dag Spicer: The Oral History of Lee Lorenzen, with Interviewer Dag Spicer, was recorded on May 24, 2017 in Mountain View, California at the Computer History Museum. We're here with Lee Lorenzen, a computer pioneer who worked on some very interesting projects in the 1970s, 1980s, and 1990s. We'll be exploring some of those topics today. Welcome, Lee. This is part of the Software Industry SIG's ongoing oral history project,

Lee Lorenzen: Great to be here and hopefully still working on interesting products. <laughs>

Early Family Life and Schooling

Spicer: Thank you. Let's start at the beginning. Tell us a bit about your younger years: where you grew up, if you have any siblings, and what your parents did—those kinds of things.

Lorenzen: My dad was a weatherman in the Air Force, so about every two to three years we moved to a different location. I had two older brothers, five and six years older than me, and then a younger sister who was three years younger than me. I was born in Waco, Texas, in 1959. I lived there until I was between three and four years old. My dad was then stationed in Alaska at Shemya. It's a little island at the very end of the Aleutian Islands. They didn't allow families there then, so we went back to Holstein, Iowa, where my mom and dad met each other, and lived there in a very small town in the Midwest. My grandparents lived with us there as well. They were farmers.

Then my dad was stationed in Anchorage, Alaska, at Elmendorf Air Force Base, so we went from Holstein to Elmendorf Air Force Base. From Waco, Texas, to Elmendorf is a big difference.

 © 2017 Computer History Museum

We were there in 1964 for the Good Friday earthquake; it was a 9.2 on the Richter scale. No one in our family was hurt but our whole house shook, and we had to run outside and get in the car. It lasted two and half minutes. It was the world's largest earthquake in a populated area up until that time. The statistic I always remember was that 150 miles of the Alaskan coastline moved 15 feet. Digging up one scoop of dirt is hard enough, but imagine the force to do that. I think at a very early age I got the sense that even the ground was not that stable, so I think that might have had some little influence on my willingness to take the risks of doing startups later on, knowing that things could change.

In 1965 we moved from Anchorage, Alaska, to Montgomery, Alabama. We took the Alcan Highway, drove all that distance down, and arrived I think six months after Martin Luther King gave his famous speech after the march from Selma to Montgomery; it was right in the middle of the civil rights era. We lived one year off base, three years on base, and then a year off base again. During that time period, things were starting to shift in terms of civil rights. My dad went to Vietnam that last year. Then I think the Air Force felt sorry for my dad, so his next station was Hawaii. We moved to Hawaii for three years, and I was there for sixth, seventh, and eighth grade. Then we went on to Hampton, Virginia, and lived there for two years and then back to Texas and San Antonio. It was in that time period that I first encountered computing.

Spicer: Please tell us a bit about computers and maybe step back a bit. Did your dad use computers in his weather predicting work?

Lorenzen: He may have at work; I never saw any of that. The closest thing that we got to a computer was my brother Gary, who is five years older than me. He was a genius, and it was kind of like being Leonardo da Vinci's younger brother. I mean he was a tremendous artist and could fix anything; he would take apart the tape recorder and put it all back together again. He would order Heathkits, and I was his gofer. I don't know that I really understood what he was doing, but if someone needed a Phillips screwdriver, I'd run to the garage and get that. I learned not really how to do all of that but just saw how an engineer thought about things, taking things apart and putting things back together again.

When we got to Virginia, the school system there was much better than the typical school systems that we had encountered next to the Air Force bases. As part of 10th grade algebra two class they had a teletype terminal that connected to a PDP-11 somewhere, a timesharing terminal. We would learn something like the quadratic formula in class and then could go write it up as a Basic program. That's the first time I learned about Basic or anything about computing; that was my first real exposure to it.

From there we moved back to Texas. back to a not so good a school system, so they didn't have anything like that, but I was pretty much hooked on this concept of programming. I think the TRS-80 had come out in my last year of high school. I remember going into the Radio Shack. They had a TRS-80, and it had Basic running on it and you could write a little Basic

 © 2017 Computer History Museum

program. I wrote “10: print ’hello, Lee’” or something like that and then “20: go to 10.” I would just sit there and the whole screen would show “Hello, Lee.” That was the level of hacking that I could do at that time.

Spicer: Did you have any interests or hobbies besides the computer, like stamp collecting or anything like that?

Lorenzen: Yes, coin collecting. My brothers did stamp collecting when we were in Alaska and that shifted into coin collecting. We would go to the bank and get a hundred or a thousand nickels—whatever we could afford—and then go look through them all to see if there were any we didn’t have. We had those blue coin-collecting books on the shelf, so that was a big hobby. My parents were very non-musical, so we didn’t really have any records in the house; I mean we had one or two records. My brothers were somewhat into music, but music was not really a part of what we did. When we lived in Hawaii, we were outside a lot, swimming, playing basketball, and stuff like that. My brothers were great basketball players; they won the state championship in Hawaii, so my main hobby as a kid was playing basketball.

Spicer: Tell us a bit about high school and then into university.

Lorenzen: When we lived in Virginia, there was a large school, and they had a very good basketball team. I couldn’t even make the basketball team as a freshman. My friends were all playing soccer so I played soccer instead and was a goalie. I wore glasses and had braces; I was basically a classic nerd during that time period.

I had contacts by the time we moved to San Antonio and the braces came off. Texas is a football state, not a basketball state, so I was a star basketball player as a junior. I made the basketball team. I would have been on the varsity if they’d allowed that, so everything sort of shifted. I went from being more nerdy to more outgoing.

My dad was in the Air Force, one brother had gone into the Air Force through ROTC (Reserve Officers’ Training Corps) at Auburn, and the other brother was at the Air Force Academy. My whole goal was to get into the Air Force Academy and tag along after my brothers. I was orienting my high school years to get straight A’s and be a super achiever so that I could get into the Air Force Academy. That was my goal.

Spicer: And did you?

Lorenzen: I did get into the Air Force Academy, and I thought that was exactly what I wanted to do in life. When I got there in my very first year, I took a computer science course. That was the first time I could really learn to program with someone teaching me, not like the thing I did in 10th grade, just learning Basic.

 © 2017 Computer History Museum

I fell in love with programming, and it was clear that that was what I wanted to do. I didn't really need to be in the Air Force, and I didn't like the Air Force Academy experience. It was a very lonely time period. It was very clear to me that I wanted to work on computers, so I was a computer science and math double major and working hard.

I was right at that transition point where they had a Burroughs 6700 computer. We programmed in Algol. They had the classic cold room with the glass. and you submitted your card deck; it was still punched cards then. You filled out your punched cards and then you got your deck. You gave it to the guy behind the glass, and he would run your program and you'd get this big printout of all your bugs. Then you fixed your bugs.

Microcomputers were just coming out. I still remember it was a classic Air Force thing: they issued you an 8.5-inch floppy disk to boot the computer. I had never touched a computer in my life up until that time. except the TRS-80. At the Air Force Academy, you'd go down into the basement of the lab, and the board said, "Insert disk. Press boot." It was just very simple. You took your floppy disk, stuck it in, and turned the computer on. I remember to this day that that was the first time I heard computing happening; you would do like a directory or something and then the disk would whir and click, and it would find the file and open up that file. I had this mental image of the Flintstones, where the bird was inside pecking out the paper or whatever. It was very different from the Burroughs 6700 where everything was in a clean room basically, and you couldn't really touch the computer. I really enjoyed that.

Decision to Leave the Air Force and First Computing Job

Lorenzen: Most of the programming was in Pascal, and by the middle of my junior year, it had not really improved. I had done well in school; I had straight A's except one B in wrestling—that was the only blemish on my attempt to get straight A's in college. I was very homesick and decided I didn't want to be at the Air Force Academy. I loved working on computers, and I figured I could do that in the civilian world; I don't have to do that just in the Air Force. Again, my dad was in the Air Force, both my brothers were in the Air Force—they could have disowned me. I had heard stories about dads telling their sons, "If you quit the academy, don't come home." But my parents were very welcoming, so I made the big decision and went back to San Antonio. I quit in January 1980 and had six months before the next semester started at some civilian college.

I applied to different places, and Southern Methodist University (SMU) took all of my credits from the Air Force Academy, except for the one B; I left that out. I had plenty of PE credits so I just said, "Don't take that wrestling credit." Because the core curriculum at the academy was so strong, all I had left was computer science courses. I had to talk the guys at SMU into letting me take six computer science courses each semester. They said, "You're not going to be able to do that," and I said, "That's all I have left so let me try."

 © 2017 Computer History Museum

I knew that was coming up in 6 months, so I got a job. I looked in the phone book to see if there was any kind of computer job. One guy named John Roland ran a computer store.

This is before computer stores really existed. This was the era of the Apple II, the Exidy Sorcerer, the TI-99/4, S-100 bus systems, CP/M-based systems. Roland owned a boarding house/hotel in downtown San Antonio, and he had turned the lobby into a computer store because he was a computer hobbyist. When I walked in, he said, “Hey, I need- help so I’m-going to hire you. It’s two bucks an hour.” I was not really very mechanically inclined. A Centronics printer cable has 60 connections that you have to do line to line; you had to test each line through. I’d work for three days to get one printer cable made, so I was not very good at that, but what I was good at was selling.

This was heaven for me because I think he had every issue of *Byte* magazine that had ever been issued on his shelves. He had all these different types of computers. He didn’t sell that much, but he liked just having them. I read the CP/M manuals. A lot of the programs you had to load stuff up on cassette tape, that’s how you got it into the computers. This was almost pre-floppy for a lot of those machines.

I landed basically in heaven there for the six months before I went to SMU. I worked on all these computers and read about them all day long. During that time period, VisiCalc came out for the Apple II. That was a seminal event in my life because I had seen for the first time the power of a killer app to not only sell the software but the computer underneath it, the operating system, and everything that it pulled along with that. No one knew what a spreadsheet was, so when people came in, I’d say, “Do you have any ledgers or worksheets or anything with rows and columns that you’d use in this?” A lot of them they had inventory-type things.

I said, “Okay. Bring me in some of those sheets.” They’d bring them in, and I’d spend a couple hours putting them in a spreadsheet. When they came back in to see it again, I would say, “Here’s your data all lined up. Now what happens if you do the math wrong or this number changes?” They said, “We have to get out the White-Out and change this and this, and it cascades over three sheets.” I said, “It doesn’t work that way anymore. There’s something called replicate or recalculate where you just change that figure and then everything it affects changes.” You could see it.

The instant that recalculate thing happened, and I hit print again, they got a perfectly printed-out version of their worksheet. They wanted the software; they wanted the hardware. It almost didn’t matter what it cost. It sounds arrogant but Lorenzen’s first law of selling software was, “Show the customer their own data inside of your application. Then you’ll sell your app and you’ll sell everything else too.” A lot of the future work that I did started with that first step: How do we get the person that we want to sell to to buy whatever we’re selling? We want them to see their own data in our application. When that works, they’ll buy it.

 © 2017 Computer History Museum

Spicer: That's a wonderful law.

Lorenzen: It works.

Spicer: It makes a lot of sense.

Southern Methodist University

Lorenzen: I was there for six months and then I started up at SMU and I loved that. I had a little garage apartment off campus, and I would sit in my garage apartment writing code. We didn't have a computer so I would write it out on a yellow pad and then go to the computer center. By that time, we had a PDP-11 that the people in the computer science department used. We had terminals, and you could actually write the code in an editor; I think Pico was the editor we used at that time. I met some great guys who were also in the computer science department. Then I got a job to help pay for school at another company. They also had a PDP-11, and they did business software for other people. They used something called Dibol, which was kind of like DEC's version of Cobol. I became a Dibol programmer as my day job, and then all my computer science projects were done in whatever they used.

Spicer: At this time, you'd pretty well moved away from cards and paper computing to interactive computing, right?

Lorenzen: Using the PDP-11 was the first time I had sat in front of a terminal. Maybe the last years at the Air Force Academy they had a terminal that you could edit your cards in, but you still had to create a punch deck, so you could kind of edit it on a terminal but you couldn't submit it that way. You then hit print, and it created a deck. Then you took the deck to the system.

Spicer: It created a physical deck?

Lorenzen: Yes, it was a physical card deck.

Spicer: That's an interesting hybrid.

Lorenzen: It was just the interim point. It was connecting the CRT (cathode ray tube) terminals up to the mainframe. they didn't really have that, plus there was no way to store your program, so the way you stored your program was the deck. You didn't want to drop your physical copy and get all the cards messed up. That was a pain.

By the time I was at SMU it was really a revelation using things like the TI-99/4 and the Exidy Sorcerer, Commodore PET, Apple II. You were editing right on the device, and the challenge

 © 2017 Computer History Museum

was what you were saving your program in. It was a cassette tape. Later on, I think we had floppies. The S-100 systems had 8.5-inch floppies.

Spicer: Was this your third year, your junior year?

Lorenzen: Yes. I quit the Academy in my third year and skipped that second half of the third-year semester at the Air Force Academy, but I had plenty of credits, so all I needed were the computer science courses at SMU. I had two semesters to finish up at SMU. My goal was basically to graduate before the people that I had gone to the Air Force Academy with. There's a competitive streak in me, I guess.

Anyway, I just took six computer science courses each semester, and I was ultimately valedictorian. Whoever was number two there I'm sure hated me because I came out of the blue, having not been at SMU, and stepped in front of him.

Spicer: Was that a full course load?

Lorenzen: Yes. No one took four computer science courses even. You would take two computer science courses maybe, and I took six at a time for two semesters. But it didn't even feel like it was hard work because I loved it.

At the Air Force Academy, you had the military requirements and the athletic requirements as well as your academic requirements. I was taking a lot of courses, but it felt like all I had to do was to program and sleep; that was easy. I didn't have a social life during college. I had met my wife during that time at the computer store. She worked as a bank teller. I would go deposit my 20 bucks or whatever in the bank, and she was one of the tellers. I just wanted to graduate so we could get married and I could start my life. That's how I looked at it.

Spicer: Would you say the Academy was considerably more demanding?

Lorenzen: Way more demanding, yes. One of the things that was a big advantage is it teaches you to prioritize things and to be able to manage a big course load and a lot of work. One of the things that they taught people that were having trouble at the Air Force Academy was how much interruptions kill your evening, so people would have a timer on their desks. If someone came to talk, "Hey, what's happening?" then you had to press the timer. You'd see that three or four hours of study time at night turned out to be 30 minutes of actual uninterrupted time. You got really good at holding a lot in your head and just kind of cranking stuff out. With six computer science courses I had individual projects in each, plus I had my work at the computer place where I was writing programs in Dibol. It was a lot of work but I loved it, so it didn't even feel like work. It was just like, "Oh, cool, I get to do this now." There was pressure

 © 2017 Computer History Museum

because I was trying to get all A's so it wasn't enough just to get through the courses. I had to get an A.

Joe and Kathy Lynn were two of my professors there; they were amazing. They were husband and wife, and I really liked them; they taught me a lot. Compilers was one of the hardest courses. You could either write a compiler that would target the PDP-11, or you could make up a new language and then your compiler could compile it to your new language. Then you had to write an interpreter that would interpret the bytecodes that basically you'd created. That was the approach that I took. I was always looking for the easiest way.

Spicer: Were there any other courses that you particularly enjoyed or that were really difficult?

Early Computer Users and Business Applications

Lorenzen: Database design was the one course where I got some introduction to user interfaces. There was something called SQL and Query by Example, QBE, so there were these things where people were trying to make using the computer more accessible to business people. I had seen VisiCalc, and to me it was like, "Look at VisiCalc. That's taking this thing that everyone has to do, and they've made it so that everyone can program."

That was a big influence. How can we make it so everyone can program? With VisiCalc, if you could write a formula in a cell you could program and you could solve so many different applications because it was a horizontal application. At that time, you got an app for writing. Balancing your checkbook was a classic example. There was an app for storing recipes or whatever. These were classic home apps that people made up, but with VisiCalc anyone could do it.

During that time period, my next-door neighbor was my first customer. I created a company to do computer spreadsheets for inventory tracking. He worked at Haggar, so he had this huge problem: he had to go around to all the stores in San Antonio and count how many pants had been sold. Then he put those numbers of pants into the spreadsheet with sizes and styles so that when the stores would come around and reorder, he could help them reorder.

He was doing all of this on big sheets of ledger paper with lines drawn and the store name, the number of pants, the number of sizes, and the counts. If he ever got a count wrong then he had to redo it all—I saw the White-Out right there. I showed him what he could do, so we created a sales-tracking program.

 © 2017 Computer History Museum

Eventually, Haggar bought that not just for him but for other salespeople. That was the first startup company that I had during that six-month time period. It was another way to make some money while I was getting ready to go to school.

Spicer: What were the computers of that era? The TRS-80 came out in 1977, and the Apple II and the PET all came out that same year.

Lorenzen: My last year of high school.

Spicer: Interestingly, if you look at most of the advertising from then. it makes this assumption that the buyer is a programmer, unlike today. Today, we're consumers, more in the consumer electronics market sense. Back then the audience they had in mind seemed to be someone who would enjoy programming their own applications, which is unusual for a consumer product.

Lorenzen: I think there was a wave of advertising a little bit before that where they said, "Track your recipes. Balance your checkbook." There'd be some basic app that you would get, but they figured out that no one really wanted to buy this whatever-it-cost computer to just do their checkbook. It sounds like a good idea, but then I think the hobbyist was definitely there. People would get these books that had Basic programs in them and just copy them from the book and then "Oh, cool, I made a game. I made something." That was fun for people that had, one way or another, some taste for programming.

Spicer: Certainly, coming in on the game channel would appeal to youngsters very much. They had the time to enter 10 pages of wall-to-wall Basic code.

Lorenzen: I think that there has also always been this sort of older hobbyist-type guy who maybe through the Air Force or maybe working in banking or something had done some programming somewhere. He could sit at home in the evening with his own computer. It wasn't just 15-year olds programming; it was probably 25, 35, or even 45-year olds buying those early systems. That's the sense that I got.

First Job at Xerox

Spicer: Did you have your own PC?

Lorenzen: No. The first time I had a PC of any kind was when I graduated from SMU. Xerox was one of the companies that had campus interviews, and they had an office products division in downtown Dallas, so I interviewed. I had heard a little bit about Xerox, but I didn't really understand what Xerox meant, and I certainly didn't understand what Xerox PARC meant. I was just happy to get a job. I think my salary was $25,000, which was a massive amount of money. I

 © 2017 Computer History Museum

went from having no money to the 2 bucks an hour or whatever I was getting for the consulting work. It was like, “Cool. We can get married. We can have an apartment. We can start a life.”

Spicer: This was right after you graduated?

Lorenzen: Yes. I graduated in May 1981 and I was married June 6, 1981. We didn’t really have a honeymoon. We went to my wife’s family reunion. My wife’s mom is the youngest of 16 kids in South Dakota, so that was our honeymoon. We went to the Fourth of July family reunion that they had every year in South Dakota.

Anyway, when we got back from that, I went to work at Xerox. I remember to this day the feeling of walking into the Xerox Mockingbird Tower’s —nine or 12 stories or whatever. Our office was on the fourth floor and then up on the fifth floor they had labs. First of all, they had these cool word-processing terminals that were portrait, black text on a white screen. They had something called a CAT, not a mouse but a CAT; it looked like an ashtray surface on the side of the keyboard. CAT stands for cathode something or whatever. You moved your finger on it, and it moved the pointer around on the screen. They called that a CAT because —it was obviously in contrast to a mouse.

Spicer: It sounds like a touchpad.

Lorenzen: It was like a touchpad that was set off to the side of the keyboard. You kind of moved things around; it almost worked like a trackball. You actually moved a pointer with it more like a trackball, but it was just a flat surface that your finger could touch.

In any case, that was cool in and of itself. I’d never really had anything with which I could write a perfectly formatted word-processing document. It was WYSIWYG on the screen. It didn’t have much in the way of graphics, some graphics, but certainly for doing Tech Pubs or the kind of proposal papers that you’d have to do at a company like Xerox. It was awesome.

Spicer: What computer was this?

Lorenzen: This was on an 8085; that is what the Xerox 860 ran on. It ran some proprietary OS that they had and it was their own native software. It was hard coded for word processing; I mean that’s all you could do with it. It kind of competed with the Lanier and CPT systems and those types of things.

One of the reasons I think I got hired was that Xerox had just released the Xerox 820, which was a CP/M-based kind of standard Z80 system. There was some company called Balcones that they had licensed it from, and they had just released it. This was six months before the IBM PC came out. They thought, “Okay, we’re going to start getting into this computing world,” and

 © 2017 Computer History Museum

that was the system, but then what they gave you was this disk cartridge; it was a 14-inch platter. It had 2.5 megabytes of hard drive space on it. It said, “This is your email system and where you store your e-mail. Go up to the fifth floor, and there’s something called an Alto. You’ll stick this in the Alto, and then you can boot up the Alto.” Probably 18 months before, I was at the Air Force Academy and they give me an 8.5-inch floppy. I booted up this thing and I got an 80 × 24 kind of CRT type and a “C>” prompt.

Now 18 months later they give me this 2.5 megabytes of hard drive space. I put it in, and I saw the future basically with the Alto. It was like, “Oh my gosh.” It was Ethernet, so you had email and all these things I had never really even encountered before.

Even further on there was another part of that floor where they had the Star. which was just shipping right then because the Alto was really 10-year-old technology by the time I was first encountering it, so it felt like magic being at Xerox and the level of what they were doing. I just wanted to figure out how we could take that and put it on these personal computers that I knew from the computer store.

Early Work on Graphical User Interfaces at Xerox

Spicer: Tell us about your impressions of seeing a graphical user interface for the first time. Did you think it was a big deal?

Lorenzen: Oh, yes, it was so clear it was a big deal. Everything about it was different, the fact that there was a mouse to move around and click on things. That opened up the whole screen real estate.

Up until that point, the only part of the screen you needed was the one line you were editing. It was nice to see the history of all the stuff you edited, but you weren’t really going back in documents and moving the cursor up. If you wanted to get to some part of the page, then you would type some command in Pico or whatever to get that to be the line that you were editing. It didn’t even feel like an editor would feel, like moving over the whole screen real estate. It was more like the way a printer worked in the sense that what scrolled by just kind of scrolled by. They began to change that over time. You could scroll back and you started to get at least somewhat in a character-oriented world, kind of WYSIWYG to what your program would look like when it printed out, and just the way to get random access to the screen with the mouse.

They had something called the Fitts’s law—I think that’s the right term; basically, the mouse became an extension of your arm. It doesn’t feel like, “Now I’ve got to think about moving my hand to the mouse and moving it somewhere and clicking on something.” You’re just visually seeing this pointer, and then you’re just doing this thing with your arm. You sort of forget about

 © 2017 Computer History Museum

the mechanics of that, so that made it supernatural. Well, not exactly supernatural in that way, but very natural in the sense that you could get to anything you wanted on a screen.

Then you had Smalltalk on that and that had icons and menus and cut, copy, paste. Never heard of that before. I mean you just deleted things, and they went away. The concept of a clipboard for moving things was new. There were two things happening at once. You might have your email program up, and you might have an editor up, and you could cut something out and move it from one to the other, so the multitasking piece was there. All of these things I had not seen before, so I was basically blown away. As they say in "Jerry Maguire," they had me at hello. <laughs>

Star

Spicer: That sounds good. I want to talk about the Lone Star, but I also don't want to skip over anything before then that we should talk about. Should we talk about the Star a little bit?

Lorenzen: I think the Star's an amazing story, but it's not really my story. I was just in awe of what they had built, and it was definitely influential. I saw multiple windows on a desktop overlapping with icons, all of these things. It was so clear that was the future of computing. To me it was like, "Why isn't Xerox doing more with this? Let's ship this thing."

Xerox had come out with the Xerox 820, the 8-bit system, and there was going to be a 16-bit follow-on. There was a big debate inside as to were they going to go 68000 or 8086? The 68000 program-type hardware was something called Shark, and the 8086-based hardware was something called Early Bird. For whatever reason, the Xerox 820 was called the Worm. The Worm came out first and then Early Bird eats the Worm; that's why they nicknamed it that. Then to the 68000 guys the Shark eats everything; that was the mentality.

There was this debate: Are we going to do an 8086-based follow-on or a 68000-based follow-on? Our group had access to the 8086 one. It ran CP/M 86. I had seen CP/M all the way back to the computer store days, so I kind of knew enough about that. I had learned how to program in Pascal, and Digital Research had Pascal/MT+. The hardware was connected to a bitmap 640 × 400 display, and it was a black on white display. It was like a half portrait basically; it wasn't a full-size portrait display. It was just kind of sitting there in the lab, and I said, "What can we do to show off what we should build next?" My whole view of life was the personal computer: Yes; Star user interface: yes. If we could meld those two things together, then we would have something successful. I got permission sort of; it was a skunk works thing. No one said, "Go work on it." They just kind of didn't know what I was working on.

Spicer: I was just going to ask you: How did you get permission for that?

 © 2017 Computer History Museum

Lorenzen: We had other projects that we were working on. We had done some utilities for the Xerox 820, and there was an upgrade to the Xerox 860 that was coming out. They were going to put CP/M on the Xerox 860, and I had gotten a reputation for being a little arrogant, I guess. I don't know where that came from. I found out later when I was leaving Xerox, because one of the managers told me this as I was going. They gave me a task to do: they wanted CP/M-80 on the Xerox 860. There was no cursor that would flash that would tell you where you were as you were editing. You needed that to do a CP/M implementation, because you had to know where your typing was on the screen, but not where the character was. They didn't know how to solve that problem.

The managers had cooked up this thing: "Let's put Lorenzen in his place. Let's give him an impossible task to do. Then he'll fail at it, and then we can say, 'See, you're not so hot'" This was a really enlightened management technique. This is the fun thing about doing oral history. You get to be the hero of every story you tell. <laughter>

I'll tell some more embarrassing stories later, but in this case, they gave me the task to do, and one of the features of the thing was that you could do subscript. There was an underline character, and no one had ever thought of subscripting the underline character. It put it below the character cell, and it solved this problem. In a week or two I saw this thing that they didn't think was going to be solvable in six weeks. Basically, they didn't know what to do with me. I could keep up with what they were wanting me to work on, writing proposals or whatever, and then still work on what I wanted to work on, which was getting the Early Bird hardware to look as much like a Star as possible. That's where the Lone Star project came into being.

Lone Star

Spicer: Can you tell us a bit about Lone Star's main features? I know you listed 10 in your video. We don't need to go through all that. Maybe give the top 3.

Lorenzen: The people that were around us at Office Products Division were skeptical of the fact that these tiny little 5 MHz 8086 CPUs could have enough horsepower to run the Star user interface. Their belief was that that really wasn't possible. I mainly focused on taking the Star desktop with the menu bar at the top, the message line and icons on the desktop, Windows for the documents, and Windows for the folders, and being able to drag-and-drop those things. You could do all of those things with a dancing icon at the beginning. It's kind of like the idle state of the machine.

We called it Lone Star because we were in Dallas. There was this little bit of rivalry between the Office Products Division and what they were doing out at PARC. You know, all the cool stuff was done at PARC, so we were sort of proud of being Dallas and Lone Star. There was one other guy who worked on it with me. His name was Steve Thomas. We needed a proportionately spaced font. Xerox had built something called the Notetaker, and they ported

 © 2017 Computer History Museum

Smalltalk to it. Notetaker was basically five years before Compaq came out with a portable computer, two drives, and then a little screen, and you carried it like a suitcase. That's exactly what Notetaker was. Notetaker had already shown people that you could do some of this bitmap stuff on these low-end architectures.

Spicer: By the way, if I could interject. Our cameraman designed Notetaker.

Lorenzen: Oh, really?!

Spicer: This is Doug Fairbairn.

Lorenzen: Oh! <laughter> Well, then, I bow, and you get to be the hero of this story.

Spicer: At the Computer History Museum, you never know. We all do different things.

Lorenzen: Really? Well, then we'll have to talk afterwards. I definitely have to hear that!

I think we got the Helvetica font out of the work that was done on Notetaker and had to redo it to make it work in our system. I designed the font sort of by hand. I got the data from there, but I had to get them into my memory system. I wrote X's and zeros for each character in just a regular text editor. I think we might have been able to load the Helvetica-I or whatever, but then I wrote some software that could read that all in and make a BitBLT strike font, which is basically a chunk of memory that has all the characters in it one-by-one. Then you could just BitBLT, which is the core action that you do when you're working on these bitmap displays. You take some chunk of it; if you wanted the letter C, then you just take that chunk and BLT it from this one location to the screen.

Anyway, that's cool that you were at Notetaker. That's very cool! It's a small world.

Lone Star Demonstration Video

Spicer: Anything else we want to say about Lone Star?

Lorenzen: Lone Star was super important in my career, because my manager a couple levels up in a different part, a woman by the name of Jackie Pakerinan, went to work at Digital Research, which was in Pacific Grove, California, which was the CP/M operating system company. I built this thing on CP/M-86 with Pascal MT-Plus, which was a CP/M language product. It was on an 8086 platform, and they were getting ready to do a new version of CP/M-86, concurrent CP/M. When she went there, she recruited me out of Xerox to go to work for DRI because of this work that I'd done on Lone Star.

 © 2017 Computer History Museum

The other thing that was super important about it was Dick Matthews, who was kind of my sponsor several levels up in the organization. This is a funny story. He wanted to show this off to other parts of Xerox. You couldn't haul the hardware everywhere, so they made this video, the Lone Star demo video was used to spread the word about what we were doing at the Office Products Division. He said, "As a reward for doing this, you get to go out to PARC, because probably you should be doing stuff out there, or maybe you should at least get to meet those guys." I was thinking, "This is awesome! I get to go to PARC!"

So, I went there and got to show the video to maybe 40 or 50 people that came to that meeting at PARC. It was really funny, because when the menu popped down, the pop-down menu drew really quick. They were like, "How did you do that?" These were the experts, you know? I said, "Well, we used a special 8-bit wide character font, so everything is on a byte boundary." That BLTing was super-fast for those characters on the menu. They said, "Well, that's kind of cheating!" I said, "Well, whatever."

There's another quick funny story about that trip. It was like a three-day trip. The first day, in the Office Products Division in Dallas, all the engineers wore something like I'm wearing now, but with a tie. Then if you were going to a big meeting, you would wear a jacket. When I went out to PARC, I wore my jacket and tie on my first day. I was looking around, and there were beanbag chairs, and people were wearing Birkenstocks. It was like, "Oh, okay, this is a little bit different." The second day at PARC I didn't have the tie anymore, and I shed the jacket like five minutes into the meeting Then the third day at PARC I was wearing jeans or something, instead of my khakis. They were teasing me; they said, "If you could just stay one more day, you might be wearing shorts and Birkenstocks with us." I was definitely figuring, "Oh, I need to get to the West Coast!"

Spicer: That's great. What was the reaction that you got from these people about Lone Star?

Lorenzen: The last line in Lone Star, it was almost like a call to action.

Spicer: I remember it well.

Lorenzen: It basically said, "Look, we can use this to tap into all these super smart engineers throughout Xerox." I was new to all this stuff. There'd been people that'd been doing this for 10 to 15 years that were really good at it. It said, " We could just give them this as a tool, and they could build apps on it, and we would be so far ahead of everyone. Let's do that, and this will get us out. We can sell more of these than we can Stars, but it'll be a stepping stone to what comes next."

 © 2017 Computer History Museum

It was really an impassioned plea: "Let's get this thing out there." But it had a word of caution at the end: If we don't do it, someone else will. We'd already heard Apple was doing something and Lisa was going to be coming out. They knew at PARC, because they were plugged into the community; they probably knew everything that was happening, and there was just a lot of head nodding. The Notetaker had shown the same thing however many years before, and now here we were again.

I think there was almost this resignation in the room, and a lot of people were plotting their exits from that room too in terms of, "I'm going to go work at Apple" or "I'm going to go work at Microsoft." Leo Nikora was one person who ultimately went to Microsoft, and later on in the story, I did some stuff with Microsoft. I think he was someone who told people, "There's another guy who's kind of doing this user interface stuff that you might want to look at."

Maybe it's just an optimistic view of life, but I thought I had impressed them. That was kind of my goal as an engineer, but at the same time not impress them like they'd never seen it before. I wanted to impress them more like, "Oh, my gosh. If someone just out of college can sit down in six weeks and do something that looks like this, then a lot of people can probably do this. It's not going to be our own little private domain like it had been for 10 years before that."

Xerox's Patents, Licensing, and Commercialization

Spicer: Right, right. Do you have any thoughts on why? There's that cruelly titled book, *Fumbling the Future*. Is that fair? Or is it fair to criticize Xerox for not commercializing more of its inventions?

Lorenzen: I think that title is unfair. It should be "Inventing the Future, and then Fumbling It." Xerox should get massive credit for first inventing it. Then, did they really fumble it? It became the future. The only thing they fumbled, if you will, is monetizing it. I've read that book and others, but I put the most credence into the fact that Xerox had been forced to divest itself of all of its copier patents because of anti-trust issues. Corporate at Xerox basically looked at all of this effort to patent things and copyright things and all of that stuff and what it had really gotten them at the end of the day. There was a judgment and consent decree or something like where they had to basically give all of the Japanese companies access to the crown jewels of all these patents. From that point on, they had massive competition, and they didn't really get protection from it.

While that's happening, they're inventing the future, and there could have been a whole series of just the fundamental patents and potentially a licensing model or other things built on that that they would have made royalties from for many years to come. Yesterday, we were sitting there with John Warnock and Chuck Geschke, and if you think about all the wealth that Adobe created, that could have gone into their coffers. If they'd managed the Apple relationship differently, they could have had a piece of that.

 © 2017 Computer History Museum

I think there were just a lot of people around who saw it as like having a very wealthy neighbor with no doors or locks. Everyone in the neighborhood just kind of decided, "Oh, this might be a useful place to borrow some stuff from." I was certainly in that case when I went to Digital Research. We weren't paying Xerox anything, but everything I'd learned there got repurposed at Digital Research, and obviously there were others in other companies doing the same thing..

I think that in terms of changing the world, they accomplished that goal. I don't think it's completely fair to just focus on the fumble and not the accomplishment.

WYSIWYG and Desktop Applications

Spicer: That's a great perspective. What do you think of the “Office of the Future,” and the idea that there will be a paperless office? That would seem to conflict directly with Xerox's business, which is generating paper.

Lorenzen: Yes, we had some internal memos that went around and kind of talked about the Office of the Future, the paperless office, and all of that stuff. I think giving people the ability to create paper at a faster rate, that's what everyone sort of missed when you enable publishing. When you enable more people than just businesses to create a flyer, you're going to have a lot more flyers.

Spicer: How crucial was WYSIWYG to that success?

Lorenzen: I think WYSIWYG was fundamental to it. The mental picture of the page that you have to carry in your head when you're not doing it with WYSIWYG is a higher order of mental function. Once you had a WYSIWYG, "Okay, I can see what I'm operating on." It also saved paper, really, because you didn't have to print it out to check what was going to be. Then you have to do another one. You'd iterate. There were batch formatting systems in that time period for doing long document stuff. If you didn't get it right, you didn't really see that until you printed out your copy. Then you saw, "Oh, there's something the matter with the footer. Got to do this over here differently."

Spicer: I remember using Word Star and the use of control characters and so on was really irritating. It made a mess on your screen.

Lorenzen: Yes, the document didn't look that pretty, and it did build a huge degree of loyalty in a certain kind of user. I know William F. Buckley used Word Star until his dying day because he could go so fast with those control characters.

 © 2017 Computer History Museum

Move to Digital Research and GEM

Spicer: Tell us about how you moved from Xerox to DRI.

Lorenzen: My wife was pregnant with our first son, so this was 34 years ago. This was 1982. It was December in Dallas, and it was cold. There was not a leaf on any tree. There aren't that many trees anyway, and there wasn't a leaf on any tree. We flew out here, and we landed in San Francisco. We had to take this little puddle jumper from San Francisco, and my wife said, "What is that?" She didn't want to get in that plane. Anyway, we made it into the Monterey Airport, and from the Monterey Airport into Pacific Grove, you drive through the forest right by Pebble Beach. It's amazingly beautiful. You fly into the airport and then you drive. You come over this rise as you're driving in on Highway 68 into Pacific Grove, and then the ocean opens up in front of you. These are not things that you see in Dallas in the winter. The trees are beautiful, the ocean's beautiful, and it's Digital Research, so it's the CP/M place I had known about for a couple years now. I was totally sold.

I told the guy that was recruiting us, "Julie's pregnant, and she loves Texas. She's been there a long time, and she doesn't necessarily want to leave, so the real sales job has to be on Julie." The two other engineers that were my hosts while I was there at Digital Research, their wives were pregnant as well. There was instant bonding among these three pregnant women. By the end of the first evening of going out with them, we could see ourselves there. I accepted the job, and then we actually moved there in January.

Gary Kildall went through the interview process. That was interesting, because it was very cordial; it was a very laidback environment. They kind of knew I'd worked on this. I think I might have had the videotape, so I think that I was able to show them the Lone Star stuff. The very first scene in the videotape says, "Xerox Private Data." <laughs>

Spicer: I noticed that. It still does.

Lorenzen: Exactly. Anyway, I wasn't worried about that, I guess. They knew that I was going to work in that area, but they didn't have a graphical user interface project at that point in time. They had concurrent CP/M, and they wanted to have a character-oriented windowing system. Concurrent CP/M allowed multitasking, so you could have your word processor, your Multiplan spreadsheet. You could have four or five different apps running. Then what we did in the character-oriented windowing system was just like a normal windowing system. You could have a little piece of the spreadsheet showing, a little piece of the word processor showing, and they overlapped one another. That was the first product that I worked on at Digital Research.

VisiCorp was working on the operating environment Visi On, which was going to be their answer to a graphical user interface. When I went and saw it, I said, "Oh, we could do better." We had

 © 2017 Computer History Museum

an Apple Lisa, and it was a 68K-based system. That was awesome that they had that at Digital Research. I said, "Let's take concurrent CP/M." They had a 68K version running on the Lisa, but they didn't have any user interface. So, I said, "Let me do the user interface for it." I don't think I used the same code, but I basically redid everything to create something called Crystal, which was the Lone Star type thing, but running on this Lisa hardware. It had a mouse, and it had a bitmapped display, but it ran on CP/M 68K.

What was weird about it was that, because it was running on the Lisa, anyone who saw it thought, "You're using some magic Lisa stuff in the ROM to do the bit stuff or to manage the mouse." We weren't really doing that, but it did make a great demo. We could certainly show the things that we wanted to show like multitasking, which no one really had at that time. The Lisa was somewhat multitasking I guess, but not in the way that we could do it with multiple windows.

That's what Digital Research focused on, their multitasking operating system at the time. We got that running. The IBM PC had probably come out by that time. They said, "Can you make it work on the PC hardware" because they had basically concurrent CP/M-86 running on the PC hardware. Then we ported everything over and got it working on that. Then people started to say, "This could be a real project inside of Digital Research."

Spicer: Throughout, from Lone Star to this implementation that became GEM, how often did you hit your head against memory barriers in terms of having enough to do it fast or properly?

Lorenzen: I have a quick aside on the name. IBM was working on something that we had heard about called Glass, which was supposed to be a graphical user interface that ran on IBM hardware. This was before Windows. We said crystal is the purest form of glass, so Crystal was going to be better than Glass, that was the mindset behind that name. We couldn't get the Crystal name for whatever reason, so I came up with an acronym—that's my Air Force Training—Graphic Environment Manager, or GEM. A gem is a higher form of crystal. You know, it's really a valuable crystal. Anyway, so that was GEM.

GEM and Step-aside Mode

Lorenzen: Memory limitations, there's two parts. The original GEM spec was that it could work on a two-floppy system, no hard drive. It would work on the IBM PC or run on top of DOS, because we knew we weren't the dominant character-oriented OS. Microsoft had beaten Digital Research to becoming the OS on the IBM PC. We were always in catch-up mode: "How are we going to get back into the game?" That was the mindset at Digital Research. If we could build a Graphical Environment Manager on top of DOS, running on the IBM PC, then that would give us an advantage. We had to make it work in what was basically a 640 KB machine. The character-oriented OS at that time, DOS, probably was a 24K or 28K, something like that. We had either

 © 2017 Computer History Museum

512K or 640K depending upon the version of the IBM PC. We had the rest of that to use. We had to get our app in the desktop. We had something called step-aside mode. That's kind of how we did it; managing the desktop was probably like a 64K app, or 128K app. to just launch the programs. But then when you launched programs, you didn't really need the desktop anymore. We could call it a step-aside mode for larger apps, if you were just going to do word processing.

There were some apps that could run on top like calculators that were little accessory apps. Step-aside mode meant that we got all of our stuff out of memory, and then we called the app and got it loaded in. Then when it exited, we could get ourselves loaded back in. That's why I call it the step-aside mode. That was one way that we worked around the memory limitation.

In the GEM team there was someone who did Paint, someone who did Draw, and someone did a Graphing program. Eventually we licensed someone to do a word processor, GEM Write from Camilo Wilson of Lifetree Software. We had those four applications, and then I worked on the desktop, and the underlying OS piece. It was a pretty small team, and we basically got it to the point where it shipped in late 1984.

Spicer: How did that relate to Windows at the time?

Lorenzen: Windows 1.0 was already out, but struggling.

Spicer: Barely useable.

Future Computing Forum in 1984

Lorenzen: Barely, yes. Not really useful at all. During that time period, before it shipped, I was the user interface guy at Digital Research. There was a Future Computing Forum in downtown San Francisco where they were going to have a woman named Therese Myers running Quarterdeck, which was DESQview. Then they had someone, probably from VisiCorp, who was going to talk about Visi On. And they had Steve Ballmer, who was going to talk about Windows. I was going to talk about GEM and what Digital Research was doing. I don't think we'd announced the GEM title yet.

In any case, you know, who am I? I ended up in San Francisco. They put me up in some hotel, and a limo came and picked me up. I'm like two years out of the Air Force Academy, and now there's some limo picking me up to take me to this venue to be one of the speakers. I was really impressed by that. I get in there, and everyone wasn’t really showing demos, but they had screenshots or something like that that you could see.

 © 2017 Computer History Museum

I think I did all my presentation on the Lisa. <laughs> Well, we didn't show it on Lisa, but that's how I got my printouts that they used as overheads. Anyway, we had a little debate about whether it was going to be overlapping Windows as the standard or tiled Windows. Microsoft was pushing tiled Windows, because that was good for small screens. We were pushing for overlapping Windows because of multitasking and the importance of that.

I told a story about that that Steve Ballmer apparently appreciated. Steve had made all his points about why tiled Windows were going to be the winning thing. I said, "This is kind of like in the Amazon in South America. When they have a herd of cattle that they want to get across the river, and they're worried about the piranha, they take the oldest, weakest cow. They take him or her upstream, and they send that cow across the river first. All the piranhas in the area go and latch onto that cow, and then they're able to get the whole herd across without a problem." After telling that story, I said, "That's essentially what Microsoft is doing. Don't you guys see what's happening? They're focusing all this attention on tiled Windows. That's their sick cow, and all the press is writing about that. Look what's happening! They've already dominated the character OS; now they're going to have the graphical OS as well, and they're going to get that all the way across the river and no one's even focusing on it. Do we really want Microsoft to have control of that?"

Obviously, we thought Digital Research should have control of the graphical operating system or at least be one of the contenders there. It got a good reaction in the audience, and I was proud of that analogy; I remember it to this day. I went back to Digital Research, and like a day later, I get this call from someone at Microsoft saying, "Steve would like to have you come up to interview at Microsoft." I was like, this is disloyal, but at the same time I'm really intrigued. After a few phone calls, I decided to go up there. He showed me Windows 1.0, because I think Windows 1.0 might not have been actually shipping yet, and I don't think you could get your hands on it. They'd been in the press, and you could read about it and maybe you could order it, but I had never actually touched it. He gave me a demo of some of the stuff.

Leo Nikora, who had seen me at PARC, I think I saw him at that time. I didn't actually go through the whole interview process, because it was more just kind of getting to know you. I did get to meet Bill [Gates], and I had a great conversation with him about why they did Excel, and what they were possibly thinking. Lotus 1-2-3 was clearly dominant, so what were they thinking? I said, "What do you have? One percent of the market? You know, two percent?" Bill sat back in his office, and he goes, “One percent this year. Two percent next year. Four percent the year after that.” What he told me was really true. He said, "Eventually, there's going to be some technology curve that the industry's going to go around, and they're not going to make that turn, and we're going to make that turn. That's when we're really going to pass them."

The graphical user interface was their chance to really move beyond. They had Multiplan, but Multiplan was killed by Lotus 1-2-3. I got a sense of who Steve was and who Bill was. It was great exposure and experience, but I didn't really want to go work for them. GEM was my baby.

 © 2017 Computer History Museum

I wanted to get GEM out there, and I was still very loyal to Gary Kildall and the guys at Digital Research. It was flattering to go up there, but it wasn't like a job opportunity. I didn't really look at it that way at that time.

GEM and Apple Litigation

Lorenzen: Then we shipped GEM, and things moved along. I told a bit of this story yesterday, but all of a sudden, we got this call from Apple. "Hey, you guys have cloned the Macintosh and have stolen all this UI stuff from Macintosh." Now this is just like your earlier story about stealing from the rich neighbor. I know we were stealing, but we weren't stealing from Apple. <laughs> We were stealing from Xerox. It was very clear to me that it was not fair that Apple should be bugging Digital Research about the look and feel of GEM. I was upset about that. We said, "Let's defend this thing." Jean-Louis Gassée, Irv Rappaport, who was the legal guy at Apple, and Guy Kawasaki all came down to tell little Digital Research how we had to change everything to make it comply, so that it wouldn't be too much like the Macintosh. We sat through the meeting, and it was very clear that management had to make the changes Apple wanted us to because they didn't have the wherewithal to fight Apple.

Later, I had heard a rumor that Bill Gates had sent Apple to do this to Digital Research because there was all this Digital Research and Microsoft animosity for a whole other reason.

Now we're up to basically the end of 1985, and I was getting ready to leave Digital Research to start the next thing. I got another call from Steve Ballmer to come up and interview officially this time. "Well, if you're getting ready to start something else, why don't you just go through the full interview process, run the gauntlet and see, because maybe we'll have a better offer for you." So, I did that. And the last part of that was talking to Bill. I said, "Bill, did you do this? I heard that you sent Apple to pester Digital Research," and he owned up to it! I've always had good interactions with Bill. I know his reputation in the industry, but to me, personally, he's always been a supporter. He certainly didn't lie about it. He said, "Yes, they came over bugging me about the UI, and I said, 'Windows is kind of like a Macintosh, but what's really like a Macintosh is GEM at Digital Research. Why don't you go hassle them for a while?'" That probably bought him six months of not worrying.

The thought process was, if they establish a precedent that Apple owns the UI, and can put a company like Digital Research out of business or whatever, make them shut down their UI, then that will have proven the legal case. Then maybe Bill would listen to Steve [Jobs] about making Windows less like the Macintosh. Anyway, it was just a delay tactic on Bill's part. It was very effective, though.

At that time, I went through the whole interview process. Charles Simonyi interviewed me, and a whole bunch of other people. I eventually got a job offer and a certain number of stock options and all of that kind of stuff. I still have that letter. For a lot of my career, I was calculating if I'd

 © 2017 Computer History Museum

just taken that job offer. <laughs> You know, "Am I going to do enough things in my career to balance the scales there?" Still working on that. <laughs>

GEM OS and Desktop Publishing

Spicer: So, this is about 1985 or so?

Lorenzen: Yes, 1985 was when we had the idea of building a desktop publishing (DTP) package on top of the GEM operating system.

Spicer: Tell us about that. I know you covered a bit of it yesterday, but why DTP? How big a user base would DTP have comprised for GEM, of all the GEM users?

Lorenzen: We didn't think of it as targeting the GEM user base. That would have been a small number of people. The GEM operating system was almost like a runtime library that came along with our app. It was very much like the graphics apps that were popular, that you needed to control the graphic screen. They didn't ship with GEM or Windows or anything. They had their own native graphics library. It wasn't like you targeted the people that were using that graphics library. They wanted whatever the game was and then got the graphics library with it. At that time, we knew if we could make a great application that people would get the operating system, and they would run it on top of DOS or whatever.

The only reason we were using GEM was we knew it inside and out. It was small compared to the early version of Windows and stable and fast, so we knew that we'd have three or four years before anyone could build a desktop publishing package on Windows that would be as small and fast and operate like ours. We had seen PageMaker on the Macintosh, which was doing great! People were loving that, and we thought that was very primitive. There's a theory in user interface that you want to match real-world objects, just give a graphical version of the real-world objects. So, you have a trashcan on the desktop, and you drag items to it. You do that at a certain level just for familiarity. That's the reason why you do that. New users will think, "Oh, that must be where I throw things away."

In my opinion, you don't take all the limitations of the real-world thing that you're choosing as an icon or whatever and write those into your application. The equivalent thing for a trashcan would be that you'd actually have to carry the trash out, put it on the side street, and then bring it back in—you have to physically do all of those steps. That would be taking it too far. That, to me, was what PageMaker was. With PageMaker, the way they laid text out was they basically had something called a window shade. I didn't even know this at the time, but the way the text is laid out in newspapers and stuff is you print these galleys that are just like one column of text at a certain width. You use an X-ACTO knife to cut that out. If you want to put a picture in, you put your picture in, and then you step down from that. After you slice the paper, you run the next

 © 2017 Computer History Museum

piece, and then you go up to the next column. It is so very hand intensive, a lot of cutting and pasting, literally. That model was exactly what they did with PageMaker.

You got a text object. They called it a window shade. You grabbed the pull-down part of the window shade and went down this far, and then the text went that far. Then you could click the mouse or something, and click up, and then you'd pick the next stop. Then you'd say, "Okay, start it up here" and window shade it down here. You're doing this all by hand. If you wanted to have like a ruling line above a paragraph, like a headline or something like that, where you'd have just a couple of lines, then after you've done all this stuff, you got your ruling line tool out and just drew a line segment where you wanted it to be. That was great for that instant in time, but as soon as your text moved, or you edited something, it all reflowed around. Your ruling line was still stuck where it was before. It didn't ride with the text element. When we saw it, it was only 16 pages.

Spicer: It was limited to 16 pages?

Lorenzen: Yes, the first version was 16 pages only. We said, "We can do a lot better than this." We were much more in tune with long document publishing, like for tech support manuals. We used something called Scribe that ran on the DEC VAX [minicomputer]. You marked up your paragraphs, and it converted everything to headlines or bylines or whatever. Then you got your finished document, ready to go to press.

We said, "If we could make essentially a WYSIWYG version of Scribe, focused on these paragraph styles and how the page layout is done, then we're going to be a lot better than PageMaker." We just saw a massive opportunity. They were so tied to the physical way you did it in real life that they didn't have any innovation in terms of how you could lay out text.

Spicer: You see that a bit in the early versions of Photoshop, which have continued through. They've modeled some of the photographic tools that you would use, like dodging-and-burning a photo, for example, by putting it under the enlarger for a little bit longer. They've kept the magnifying glass and the clipper.

Lorenzen: A little too literal. It's one thing to say, "Oh, cool, I understand it." It is better obviously to do this than use a bunch of X-ACTO knives, but you miss the chance for a big innovation.

Spicer: Unfortunately, you also bring forward all the limitations of that metaphor.

Lorenzen: Yes.

 © 2017 Computer History Museum

Ventura Startup

Spicer: Tell us how did you split from DRI to start Ventura? How did that happen?

Lorenzen: Don Heiskell and I had made the decision that we were going to leave, and we knew we needed some guy to "wear the suit" and be the marketing person. That person was John Meyer, but we didn't recruit him until later. It was really the two of us. Don was my boss in the graphics group, and we said, "Let's go build some kind of app," and we decided on desktop publishing.

We knew that we couldn't write any code on any Digital Research equipment because then there might be IP issues associated with that, so we just started taking longer lunches. Over lunch we'd talk through the design issues and how we were going to do it, and we figured out a way to get out cleanly from Digital Research. We proposed that we build it as the next app on top of GEM inside of Digital Research. We anticipated that they would reject that because they didn't want to invest in apps. The Apple thing was happening, and they were feeling like, "This GEM thing isn't going to really work for us." There were a lot of reasons why they would say no to that.

When they said no to it, it opened it up to us. We said, "We still want to go do it, so can we use the GEM layer? Maybe that'll help other GEM apps be an example for that." They agreed to that, so we got a license. We rewrote the 64KB GEM layer down to 24KB, because we only needed that much for the features of it that we were going to use inside of the app. Then we got all the screen drivers and printer drivers; that was important. because those all came with the GEM execution libraries.

Next, we found John Meyer and got him to agree to join us. The three of us quit, and we raised our money.

Comdex Demo and Licensing to Xerox

Spicer: How long did it take before you had a product to ship?

Lorenzen: We quit in September 1985. Comdex was in like November 1985, so we had six weeks to get something demonstrable. Very much like the Lone Star demo, we wanted to have a killer demo that would show people what you could not do in PageMaker, which was autoflow two columns of text and then put a graphic right in the middle of it and change the size of the graphic. If you'd have done that inside of PageMaker, you would have had to reflow every little piece when the size of the graphic in the middle of these two columns of text changed. We were autowrapping around it. It was very clear to everyone that PageMaker didn't have anything like that. We also showed style sheets. We showed enough of the style sheets were working that we

 © 2017 Computer History Museum

could have a document like a resume, and just by changing the style sheets, it would look different. That again felt very different than the by-hand way of doing things. Just by changing the style sheet, you could very quickly get completely different looks to the document, and people really liked that feature.

We had a killer demo, and John Meyer contacted 35 companies—everyone who had a printer or a word processing product of any kind. We tried to get those OEMs (original equipment manufacturers) that were all going to be at Comdex to come by our suite at the Riviera Hotel. We lined them up in the hallway and did the demo.

It was very clear we were going to OEM this to somebody, so it was just a question of who. The three that were most interested and carried on the negotiations after that were Xerox, Kodak, and MicroPro with WordStar. We just managed those negotiations. The key thing then was to actually ship the product. We cut a $5 million nonrefundable advance against royalties, AAR, deal with Xerox. I remember that acronym to this day. It was my goal to be a millionaire before I was 30, and I was 28 at the time.

Anyway, we had to ship the product to get that money. My family had invested, and we were running out of money about six months in. Xerox paid us like $50,000 more to do a printer driver for one of their printers. That got us through to shipping in probably November 1986 in time for Comdex. So, it took us about a year, I would say, from when we started to having a product to ship.

Spicer: Why the name Ventura?

Lorenzen: We liked Spanish-sounding names because a lot of people in the U.S. can pronounce Spanish-sounding names because we're close to Mexico and the whole Spanish influence of that. Texas as well. The word we wanted was Ventana, which means "windows." It was going to be Ventana Software, because obviously Windows was core to everything that we were doing, but that name was already taken. Someone already had Ventana. In the same Spanish-to-English dictionary, close to Ventana was Ventura, which means "good fortune." So, we picked Ventura. There's a song "Ventura Highway," so it had that connection as well. It was really just because it was right there in the Spanish dictionary when we picked the name, and we were able to get that name.

Ventura Templates and Style Sheets

Spicer: Ventura was actually agnostic in terms of the word processor you fed it to, right?

Lorenzen: Yes, we were not sure that we could build a desktop publishing package with all proportionately spaced fonts and with all the stuff on the page that would edit fast enough to be

 © 2017 Computer History Museum

the default editor. We really didn't view ourselves as a word processor. We viewed ourselves as a publishing system that would allow you to assemble the text and graphics and long documents as well, but we weren't sure we were going to be fast enough to allow editing.

Our workaround that was that when you loaded up the software, when you were loading up all the paragraphs that were going to be part of your resume or whatever, they first come up on the screen as a single column and everything was just body text. Then you could select a paragraph and tag it to a different style, and the document started to take shape. If you tagged something as Headline, Company Name, or whatever the thing that the style sheet assigned to that, when we saved that document out, we didn't save it into some proprietary format. Rather, we saved it back into the original word processing format. If you used Word or any editor, we just embedded in that same document those paragraph tags. Then anything that we could render naturally in the word processor like underlining and bolding, we put that into the native format. We didn't screw up your document that much. Only special paragraphs that weren't body text got the tag, and the rest of the stuff was sort of embedded. This ended up being a massive advantage.

We didn't think we were going to be good enough to convince people to stop using their main editor, but it ended up being a real coup for people who wanted to do long documents and database publishing type work. They could format like one or two pages of their video catalog, let's say, and then save that out. It might have like 24 video titles and descriptions of videotapes or whatever and pictures of that, but it was all still embedded. Then they could take that to their database guys and say, "We've got 6,000 videos in our catalog. Go emit a file that looks like this with these tags in this order." Then the next time that was loaded into Ventura, you had a 300-page document all fully formatted, and you didn't have to tag by hand. It had all been tagged before it even arrived inside our system. By writing it out, you basically created a template so that people could know what they had to do and what they had to match in order to make the ultimate document look the way it should.

Spicer: You kept the format of the originating word processor when you wrote the modified file back out, right? Would you change the name so that it knew that it was a Ventura file versus a .doc file or whatever?

Lorenzen: No, the way Ventura worked was that you had this toolbar area. Your style sheet had all the tag names—headline, byline, whatever. Then it was all the files that made up the document, all the graphic files, all the text files. Those files would have extensions like .doc, .ws, or whatever. We just kept those native files until you hit Save, and we overwrote the original. That was the interface. We didn't create V1, V2, or V3. When you saved, if you had edited that text or done any markup in the text by tagging things, then the next time you saved at the Ventura level, you had written out all those files. You weren't constantly propagating new file names.

 © 2017 Computer History Museum

Spicer: That's really clever.

Lorenzen: Yes.

Ventura Competitors and Pricing

Spicer: You mentioned Adobe. Any other competitors to Ventura?

Lorenzen: We didn't view Adobe as a competitor. Adobe's Postscript was really an output device for us. Aldus was definitely the competitor.

Spicer: I'm sorry, I meant PageMaker.

Lorenzen: Aldus PageMaker. Eventually they were acquired by Adobe, so I see the connection there. At the time, they were the main competitor. Eventually, Quark Express came along, but for the first two or three years, it was really just making sure that we were way better than PageMaker. That was our goal.

Spicer: Did Xerox promote Ventura?

Lorenzen: Yes, Xerox did a great job. There were only five of us in the company. Don Heiskell was our VP of Engineering, and I was VP of R&D. Our senior software engineer was John Grant, who still works with me actually to this day in another company I'm working on. Then John Meyer was the CEO. That was the company. All the testing was done by Xerox, except for the first-level testing. They would give us bug reports, and we would work through the bugs. They did all the packaging, they did all the selling out to the dealer channel, and they did all the training of people. They had all the staff to do all of those things. There were very few companies that had the revenue per employee that we did, given that we were such a small company. We had almost no overhead. We had an office manager, but that was the extent of our overhead. Everyone else was doing something core to the product, and then Xerox did all the heavy lifting.

Spicer: What was the retail price?

Lorenzen: $895. There might have been a $695 and an $895. Shortly after the first one shipped, we came out with a Professional Edition. So, there was Ventura Publisher and then Ventura Publisher, Professional Edition. That one had some higher-end features. Maybe it had access to more memory and probably tables. The first one might not have had tables. It had some things that allowed us to charge a little bit more.

 © 2017 Computer History Museum

Spicer: Were there any large key customers or institutional sales? A company that bought a thousand copies, for example?

Lorenzen: Not that I was really plugged into. There might have been. I think it was mainly people in corporations that had long document issues that wanted more control than their markup language, Scribe-type approach gave them. Interleaf and FrameMaker were at the high end. They worked on Sun workstations and we had most of their features, but we ran on much less expensive hardware, and the software packages were less expensive. That was at one end of the spectrum, and then you had PageMaker, which was just like simple newsletter stuff. They couldn't do the kinds of documents we did, not at that time anyway.

Ventura Sale to Xerox

Spicer: Anything else you want to say about Ventura before we move on?

Lorenzen: Xerox wanted to open up a software division, or someone had talked them into it. "You've got all this revenue, like $50 million in one year coming in. It'd be more valuable if you had a software division. Let's do a software division and get some other apps." A guy convinced them of that, but they said, "We don't own Ventura." He said, "Then you need to buy it."

We went through the process of negotiating a sale, and that worked really well for us. That was the right time for us as partners, as we were kind of ready to do the next thing. We sold the company, and it was off and running to the next thing.

Just to set up sort of the next part of the story, the last step in the process was they wanted a Windows version and an OS/2 Presentation Manager version and a Mac version. They hired someone to do the Mac version, and then I did the OS/2 and Windows versions. I didn't want to write it all over again for OS/2 and Windows since I'd written that original 24 KB GEM layer and really all of our dependencies flowed through these GEM calls. We basically wrote a GEM emulator, where we emulated all the function calls on top of Windows and on top of the OS/2 Presentation Manager and then set that up. A similar thing was done for the Mac. In about two weeks. we had it working on the OS/2 and Windows platforms.

It took the other team a little longer on the Mac side, but it got me thinking about this emulation idea. This was at a time where the Mac had all the great graphic applications. We're talking 1990, so it was the Windows 3.1 timeframe. Windows was finally at a point where you could do cool graphics applications and actually make them work.

 © 2017 Computer History Museum

Altura Software and Mac-to-Windows Technology

Lorenzen: The day after we sold Ventura, I started Altura Software. Ventura was "good fortune," and Altura is "to the heights" or "to the heavens," so it's more like a focus on personal, aspirational things. I wanted to do something in database publishing because high-end catalogs were coming out with color and images. There'd be a picture of a woman in a dress or whatever, and she'd be holding a bag and her shoes. They had all these letters next to her and descriptions. So that's the purse, that's the shoes, that's the dress.

We couldn't do anything like this inside of Ventura, so I said, "Here's a chance to do database publishing or print catalogs. I'll start a company to go do that." That was far enough away from Ventura. It wasn't desktop publishing; it was more high end with print catalog publishing. I was thinking about the next thing.

While we were working on that, I knew we were going to write that app on the Macintosh, but I said I didn't want to ignore the Windows market. So, the very first thing I started working on was an emulation layer where I took the Mac API and mapped it into Windows. I'd just done that with the GEM API, and Mac was enough like GEM— <laughs> Apple was right, it was like it— that it wasn't that hard to write this Mac-to-Windows porting layer.

We were going to use ourselves for the app we were going to build, but as soon as I wrote that I said, "Oh, my gosh! There's like an opportunity here, because you have all these companies that have Mac applications that want to get to Windows." I knew how different they were. You need two different development teams, and it's a huge expense to manage two development teams. We started licensing this Mac-to-Windows technology and that became a very profitable business. That was Altura's business. Macromedia Director was ported that way.

Quick aside: One of the things that we used to do was test our Postscript output at Aptos post typography. A guy by the name of Steve Manousos had the first Linotronic west of the Mississippi. Basically, we used to test our desktop publishing output at his place. While I was there, I met these other two guys, Mark Zimmer and Tom Hedges. They had done Color Studio over on the Macintosh side and were working on something new. Ultimately, it was called Painter, this natural media Paint program. I said, "Oh, my gosh! This is the coolest." If you're around long enough, you know when something's going to be a success—like the first time I saw VisiCalc. It was just an alpha version that he showed me in his office one day, and I said, "Let's start a company. We just had this big success with Ventura, so let's start a company."

We started Fractal Design Corporation and went to market with Painter. They only knew about the Mac API and wanted to write the Mac API. I said, "No problem. I'll do the Windows version." We built the Mac-to-Windows layer so that Painter could use it. I think that was probably 1990 Comdex. We showed that in the Microsoft booth. Bill Gates mentioned it in his keynote address as one of the coolest apps at Comdex that year.

 © 2017 Computer History Museum

Every week, I would open up *MacWEEK*, and if anyone was talking about their Mac application, I would call them up on the phone and say, "Hey, do you want to get to Windows? We can get you there in six weeks' time and use the same code base. Here's how it works." That's how we built the Mac-to-Windows business.

Fractal and Painter

Spicer: For people 100 years from now, tell us what Painter does or did.

Lorenzen: Probably to this day, Painter is the most amazing app that I'd ever seen. In normal paint programs, you pick a color, and maybe you have a brush, a spray can, or something, and you can like draw with that. But what you end up creating, anyone on the planet who looks at it would say, "That was done with a computer paint program." Painter had something called the Natural Media Paint program. One of the key enabling technologies was something called the pressure-sensitive stylus. The Wacom had a pen-like device, and when you pressed down on the surface, it would register between one and 256 levels of pressure. It was no longer a mouse that had either on or off; the pixel could be on or off, and you could tell the software that the pixel should be one to 256 times on or off. It typically wasn't one pixel. It's like an area.

The other thing that they did was get a number of patents; they did all the work, and I just did the business stuff for this one. They had paper textures. Artists use all of these different kinds of paper, for example, charcoal absorbs at a different rate based on how hard you're pressing it. So, think about drawing with charcoal, and the whole way in which artists do what they do. If they press hard it's really dark, and if they press light, just a little bit of the charcoal comes out. A crayon and oil paints have the same type of thing. It was all based upon the paper textures and the pressure-sensitive stylus. When this company went public in 1992—that was my third startup after Ventura and Altura—people had never seen anything like this. When you were done, it looked like you had taken a photograph of some oil painting or charcoal painting or watercolor painting.

Another thing was very innovative about that company; we sold the first product in a paint can. The designers that we worked with came up with a couple different suggestions, but at the time, all software sold in a binder and sleeve. They looked like bound books on a shelf, essentially three-ring binders. It was very expensive, like maybe $2.50 or $4.00 to just get the binder sleeve you put all your floppies in that shipped out when you shipped your product. The designers had the idea, and we agreed, to ship it in a paint can. When you pop open the top of a paint can, it's wide enough to put a 5.25-inch floppy inside of it. The documentation and the poster had to fit in there. All the things fit in the paint can.

The beautiful thing about this was that in the retail environment it stood out from everything else. The vendors in the computer stores could make piles of the paint cans that were empty and

 © 2017 Computer History Museum

then have the real ones for sale at the top. The real win was that the can itself cost 25 cents instead of $2.50 or $4.00. It was much cheaper and distinctive. Everything about Painter was aesthetically perfect. Mark Zimmer and Tom Hedges, all to their credit. That was Fractal.

Later Startups and Investments

Lorenzen: Now I get to tell one of the embarrassing stories. If you go back to when I sold this sales tracking thing to Haggar, that was a success. Then Ventura was a success. We did Altura, the Mac-to-Win software, and that was a success. Fractal was a success. That went public; I was on the board, and I got to take a company public. I said, "This software thing is really cool! Just start a company. A couple years later, you're making a couple million dollars from that company one way or the other." So, I said, "Forget about one or two at a time. I'm going to do seven companies at a time!"

So, I started seven companies, and all of them completely failed. I got to give back to the house some of the winnings, a lot of the winnings, from all of these different companies. I definitely learned a lot of lessons. Maybe it's an apocryphal story. Someone asked Edison about the process of inventing the light bulb. They said, "How many times did it take?" He said, "I did like 996 tests." They said, "Wow! You had 996 failures before you were successful!" And he goes, "Oh, no, no, no. I had not failed. I just knew 996 ways not to make a light bulb." In essence, each of these experiments was not really a failure. It was a learning experience. I'm kind of that same way in the software world. I've started about 25 different software companies. About five of them have been successful.

I know a lot of ways not to make a successful software company. They say you learn more from your failures than your successes, and that tends to make me the resident genius in most rooms. Aside from maybe a few VCs (venture capitalists) in Silicon Valley, there are very few people who have had as many software failures as I have.

This is the way I explain it to my wife, who's been very gracious through all these years: "We're still playing with the house's money, you know?" She said, "Why can't we just stop when the revenue is at the peak?" Unfortunately, we were not smart enough to do that, so we get to give some back. That's why I'm still in the game really. To me, those were some of the early successes and then on to Catalog City, which became Shop.com, and now I am working on other things.

Historical Observations

Lorenzen: I feel a little bit like the Forest Gump of software. At various points in time in my career, I've managed to put myself in positions where I'm seeing history happen; getting to know Bill Gates and Steve Ballmer in the early days of Microsoft, actually being an employee of Xerox

 © 2017 Computer History Museum

of all places, being at PARC. I wasn't really involved in some of these things, but I was certainly on the scene observing.

Fast forward to the future. When Facebook first came out, I made this prediction that Facebook would be worth $100 billion. I took all this abuse. This was in August 2007, and I got to do early stuff on Facebook. It feels like a wonderful career, essentially. "It's a Wonderful Life" is my favorite movie because you get to sort of be in all of these places. Unlike the movie "It's a Wonderful Life." where maybe you've made an impact and people have benefited from you being on the planet, I'm not so sure that many people have benefited from me being on the planet. Maybe my family.

Spicer: Oh, c'mon! <laughs>

Lorenzen: It is very interesting to see the impact that things have had. This may go all the way back: I was never intimidated by powerful people. My dad was 6'4", and my brothers were 6'6" and 6'5".

Spicer: Wow.

Lorenzen: So, growing up, I was this little guy looking around. I'm 6'4" now.

Spicer: You're only 6'4". <laughs>

Lorenzen: Yeah, I'm 6'4" now, but my personal image of myself I think is that everyone is bigger than me. My dad was an officer, so he got all this respect in the Air Force from the people around him, but to us, he was just our dad. It's just kind of like, "So what if he was eventually a Colonel in the Air Force? He was still just my dad."

When I went to the Air Force Academy, for all these cadets and even some of the officers, when a Colonel showed up it was a big deal and everyone snapped to attention. To me, it was, "Okay, he's just a Colonel. He's not like a three-star general or something. Let's not get too impressed!"

When I was interacting over time with these heroes of mine, I was never afraid to challenge them about what they were doing, so I think you get more respect. I used to send two or three emails a year to Bill Gates for a lot of years. I would only send them when there was something going on at Microsoft that I didn't think they were doing right and that I thought I had some sort of like different angle on. I don't think that other people are like that; at that level, they have many people around them that are Yes Men and Yes Women.

Spicer: Exactly.

 © 2017 Computer History Museum

Lorenzen: If they actually get input from someone who has a contrarian view, I think they respect that more. That always worked for me. Anyway, there's a lot more history in there, but I don't know how much time we have.

Spicer: I think famous people get tired of being famous actually and being treated deferentially and constantly agreed with. It's very corrosive, I think, after a while.

Lorenzen: Yes.

Spicer: They appreciate honesty.

Lorenzen: I think that there is a feeling too that probably the most honest people in their lives are the people who knew them back when.

Spicer: Yes, before they were successful.

Lorenzen: Before they had whatever level of success.

Catalog City and Ecommerce Shopping

Spicer: Anything else you want to chat about? Current projects?

Lorenzen: Current projects! I'm happy to talk about my current project. First of all, I started Catalog City, which became Shop.com. That was one of the companies along the way. I mentioned that I was going to do a desktop publishing package for print catalogs. When the Internet was getting started, I went home one day, and our house had received 25 catalogs in a single day. I wasn't buying much from catalogs, but somehow the catalog industry decided to send 25 catalogs to us. I learned later that a print catalog arriving in your mailbox costs the catalog industry about 70 cents. In one day, the catalog industry put $14 in our mailbox. And these are paper.

Spicer: Different companies.

Lorenzen: All different companies, yes. Once you buy from one, they share the list, so everyone's sending you catalogs. So, I said, "There's got to be a better way. We can put these catalogs on the Internet." At the time, Amazon had just launched with books. This was 1996 or 1997, and they had maybe launched the year before. They were selling books through a shopping cart; you could buy any book you wanted. The web rewards authoritative selection; that's one of my rules of success on the Internet. So, they had all of the books, a million bookstores or whatever, in one location. You could buy your three books, and you put them in your cart and checked out.

 © 2017 Computer History Museum

There was a document that I had from 1990 when I was going to do this print catalog desktop publishing business that had all the print catalogs in the U.S. I looked in that catalog, and I said, "What's the biggest category of products that are sold through print catalog in the U.S?"

Books, I would not have predicted that, but I thought Amazon was doing that. What's another? There's a lot of computer catalogs, so that was another category. Someone's going to do that category. Apparel was obviously a big one, and collectibles was huge, but eBay had kind of gotten that.

I was like, "The catalog book is kind of like a guidebook to what is going to sell well online." These items were selling at a distance with nothing more than a picture and a text description, with someone managing the inventory remotely. To me that was a great predictor of what was going to happen on the Internet. When I got the 1997 version of that book, it went from something this small to about this wide. There were a lot more print catalogs for whatever reason in this later version, but the same categories still held. I had this predictive guess as to everything that was going to be successful selling online because the early adopter cases were already proving to be successful.

I wanted to start a company to basically go after online sales of everything sold from print catalogs. To me, Jeff Bezos was just like Paul Brainerd at Aldus; they were too literal in the sense they just picked this one category. Amazon was just a book cataloger with a website; that's all they really were. I said, "This is silly! Let's do this with virtual inventory." Basically, having warehouses, packing, and shipping sounded hard to me. I said, "That's already out there. Let's just take all of the print catalog inventory that exists and put it in one single website."

We created Catalog City in 1997. We wanted to have all the catalogs. Since catalogers would send you their catalog if you requested them, we sent out 5,000 requests for catalogs and said "Mail back your catalog in this." We got like 3,000 in the next two weeks. We bought like 50 scanners and scanned all the covers. Then we started scanning the pages. The idea was to sign up these print catalogers to eventually host their inventory. Because they didn't have their own websites yet, we wanted to get their inventory feeds, and then have a single shopping cart. We got a patent basically from that work, which was a multivendor virtual inventory with one cart and checkout process. A guy named Jordan Zimmerman and I got the patent. That was another thing on my bucket list was to get a patent.

In essence, the business model embodied in that patent is what ultimately is more than 50 percent of Amazon's revenue right now—not the stuff sold that they own in their own warehouses, but all the stuff that they get from third-party merchants. That is either drop-shipped from the third-party merchants or fulfilled by Amazon. When we were getting ready to launch Catalog City, I was talking to a potential investor who said, " That's a stupid idea. Microsoft will just do that."

 © 2017 Computer History Museum

For any entrepreneurs out there listening, never listen to that! “Don't do that, because Google will just do that" or "Facebook will just do that." Just go ahead and do it. You do it in a better, different way than they will, and just run as fast as you can. They may acquire you, or not, but there'll be competitors that will want you.

I said I'd heard that enough times, so I'm going to ask Bill what he thinks of this idea. So, I got a meeting with Bill and pitched him the idea. I brought that big thick catalog book and showed him. "Here's eBay with big collectibles, and Amazon did this. This is the future! Let's do this!"

He asked me one core question; Bill is known for hard questions. He said, "But you don't have inventory data. How are you going to be able to know that these orders are going to go through?" I had one fact that I kind of pulled out of the air: "Thirty-eight percent of all orders that went into print catalogs went over the phone or actually through the mail. The catalog companies had to have enough inventory for what they were showing in their print catalog, so basically, they overstocked the stuff so they would never have an out of stock problem. You didn't have to have real-time inventory feeds." I got through that question, and so Bill said, "Yes, I'll invest in your company."

I said, "What do you want?" He wanted 15 percent of company. We ended up selling him 10 percent. It was $20 million post-money evaluation, so that was the first seed round investor that we had. All the fundraising after that was easy, because it was like, "Oh, Bill's invested in this! And it's ecommerce. Blah, blah, blah!" We raised a bunch of money and, this is another embarrassing part, ultimately found a way to not be successful with Bill Gates as our seed round investor. Ultimately Amazon invested. That was a big mistake—not a mistake that Amazon invested, a mistake what I did with their investment. I think it was about 17 days after 9/11, and they put what was effectively $5 million worth of Amazon stock into us at that time. As of today, I mean, I could probably check my stock quotes.

Spicer: Oh, dear.

Lorenzen: Amazon's price per share on that date was $5.87. <laughter> Had I just margined that stock instead of the next day selling that stock, it would be like $780 million. That was my big mistake during that period of time. We brought in a VC, and I lasted about a year after the VC was there.

Then I started another company with my son, Matt. Along the way, we discovered Facebook, and then during that time period the Egyptian Revolution happened. I began to understand the idea that an operating system and a killer app don't have to just be in the software industry. You can do that with a social operating system, which was Facebook, and the killer app was the way in which they organized the people on top of that around a single Facebook page or group.

 © 2017 Computer History Museum

Current Projects and Reflection

Lorenzen: Basically, I came up with the idea of an influence operating system. That's what I'm working on right now, weaving together 15 million women around the world as shop owners who can basically represent the buying behavior of 100 of their friends, family, etc.; buy products from any U.S. merchant and have them shipped to wherever they live. The core idea is to give everyone on the planet a U.S. identity, a U.S. email address, and a payment method that works in the U.S. so that if they can't physically immigrate here, they can virtually immigrate here. That's the core idea behind ShopO.com. ShopO stands for Shop Owner, but it's my second try at ecommerce, after failing so miserably.

I joke around sometimes that the one other thing I'm working on is a time machine. The only thing I need the time machine to do is to get my voice, maybe like two minutes of my voice, back to me on a certain number of days after 9/11 and say, basically, "Margin, don't sell." I haven't figured that one out yet. I could talk for hours, but I appreciate the chance to talk here.

Spicer: I think we should wrap it up.

Lorenzen: Okay!

Spicer: Is there anything else you want to mention?

Lorenzen: I've worked with a lot of amazing people along the way and I've had many opportunities. People have had faith in me, and I joke around about all the money that I've raised and lost in all of these 20 different startups. I always say you should invest what you can afford to lose and have us stay friends. All of the people that have written these checks that haven't gotten an investment back, they've honored their side of the agreement, and I'm very appreciative for the chance to try to grow their money while living a pretty wonderful life! So, thanks.

Spicer: That's lovely! Thank you!

 © 2017 Computer History Museum