Computer Aided Design (CAD) Pioneer Workshop
Day 2 Session 5: Later Technology Development (1980-2000)

Moderated by:
David C. Brock
Burt Grad

Recorded May 2, 2023

CHM Reference number: 2023.0065

Table of Contents

David Brock: Thank you, everybody, for joining once again. It's really great to have everyone back for day 2 of our CAD Workshop. Let's open things up and get right to it for the first open discussion session on technology development, 1980 to 2000. I thought we could begin with Peter sharing for a little while about his experience, and then we will open it up to general discussion. Peter, why don't you open things up for us?

SDRC

Peter Marks: I'm probably not the person to go deep into the technology. This will be an all-hands thing, but I can maybe set the stage and put a little bit of SDRC's experience into this.

In my mind, 1980 to 2000 was the period of time that we got beyond 2D and 3D wireframe geometry. Generally, I think we really got 3D geometry in various forms that you could run other applications on. These are the two decades of really getting productive applications built on the geometry as well as just for the documentation and the rest of that. You had during that decade guys like Bob Johnson running around giving the solids gospel. You had people trying to figure it out. SDRC was a good example of this. What is NURBS? The geometry we want to use. How do we trade off rapid response time versus accurate geometry? How do we stitch these surfaces together so that we can have watertight solids and so on?

During this period, starting about 1988, I was at SDRC, and I want to talk a little bit about the culture of SDRC because I think it greatly influenced the software development. SDRC had been a client of my technical film business. They started out in this big restaurant, so they had lots of rooms. It was a good area. The bottom floor of the restaurant was a bar and a restaurant room. Every Friday the group would meet and exchange ideas. It was like a Silicon Valley company before Silicon Valley. Part of the magic of SDRC is that the company started solving problems. It had this amazing group of problem-solvers under Jack Martz. If Caterpillar wanted to have a better excavator, Oldsmobile wanted to have an engine that was smoother, or Cadillac because of the energy crisis wanted to downsize the DeVille and turn it into what was the hugely successful Seville, SDRC was the company that could help out with that. The company used its own software. It used its superb analysis, the SuperTAD fine analysis, and its Modal analysis, and their Rotordynamics software that it had picked up, and testing to solve all those problems. In a way, this was an analysis company a little bit like a firm that had design and manufacturing together. You actually saw the application of your problems and could figure out where the software wasn't working, where it was, get new ideas, and the rest of that. It was a wonderful mix.

I joined the company, and they sort of bought out what I had of an education business and decamped to Milford where they also kept the little bar inside in part of this brand-new building they built for themselves. This heritage continued.

Now it was up to the 1980s, and this is a very rough demarcation. It was mostly testing of structures and analysis of structures and the geometry of that. But the company very quickly realized that it needed a fuller suite of tools to solve problems. I came from a little bit of a cognitive psychology background as well as engineering, and I had these four learning, solving styles. We solved problems with words, analyses, and of course graphics and physical modification. I'd written up a little bit on that. SDRC realized it really needed a much fuller suite of design and graphics tools.

The culture of SDRC was set by Jack Lemon. I put a link to one of his obituaries. He was just an excellent man, a terrific leader who managed to recruit and bring in just amazing folks. Al Klosterman was the guy on the technical side who initially was the leader in understanding some of these structures but then understood who he needed to bring in to solve a problem. Bob Abarbanel came on, and he had this kind of imprecise faceted model with the exact model behind again performance. We were talking yesterday about how probably the first practice NURBS, or commercially NURBS, came when Wayne Tiller and Les Piegl came onboard.

They developed a pretty neat thing. I think Dave [Kasik] could talk to that. It developed this pretty full set of geometry tools, GeoMod. Then we also realized we needed to handle the data management side. Al talks about the weekend that he and I and Rick Briggs, who was heading up the effort, did a proposal to get a relational database that we could integrate with a CAD system, so you can manage some of that product data. And today, that is now under Siemens. SDRC tried to go through GE, but Siemens is the heir to all that technology, so there was really a set of tools that handled descriptions like bills or materials kind of language. Graphic descriptions, GeoMod, of course the modeling. Analytical descriptions, the simulation: you could actually test and see the correlation of the analysis with what was going on. I think that was very healthy at the time for the development of the software.

Maybe that's a kickoff. Thoughts, questions? Others weigh in.

<u>Advances in Data Processing and Storage</u>

Michael Payne: Let me jump in, Peter, with something completely different. In the period of time that you're talking about, this thing called Moore's law applied. Moore's law wasn’t just about memory size. It meant computing results and a lot of other things. The memory was just part of it. If you do the mathematics, roughly, the back of the envelope, and we don't have to be exact, the capability of computers, shall we say it doubled every four or five years? It was more than that. What you could do drastically changed over, let's say, a four-year period—whether that meant you could store the pretty picture inside the model so that you could retrieve it more rapidly, blah-blah-blah-blah. Doesn't matter. There was this rapid evolution of capability of every aspect.

Then you have the web on top of that. It wasn't so long ago, this is after 2000, but you could put anything you wanted on your computer, but you weren't allowed to go to the web in a big company. Now it's the other way around. Somehow your computer's a problem today, but the web isn't. I think the point is that the underlying technology caused all that to happen. I'm sure we'll hear more later about what did happen, but it's inevitable; it's not the same world in four years' time. Whether that continues or not, ad infinitum, who knows? We probably don't, but I'll shut up there and somebody can contradict me.

Marks: Let me just jump in real quick and say that that's an excellent point, maybe for the historians, I used to keep this little chart of model complexity and product complexity. If you look at a company like 3M, they had 60,000 SKUs to keep track of but relatively simple products. AMP would be another one with connectors and things like that. You look at Boeing and they might have a dozen products, but a million parts in them. Modeling started at the small corner of low-model complexity—a few primitives and a single part. From 1980 to 2000, we really expanded to both greater model complexity and incredible product complexity. Dave can tell us all about product complexity because I think that Boeing still has probably the most complex commercial product in the world.

Brock: I'm just going to jump in here and say I have John and Brad and Dave who I know all want to speak. I also want to recognize that Ken Versprille has been able to join us. Welcome, Ken.

Marks: Do want to give Ken the five minutes for his own life and stuff just so that history can be recorded.

Ken Versprille: Five minutes for my whole life?

Marks: I know, it's just done very quickly.

Brock: Actually, I thought that because Ken joined the workshop just in the section where we're really talking about technology developments in the 1980 to 2000 period, it would be a good place to give everybody in the workshop an opportunity to kind of talk at a little bit of length about their contributions and their personal experiences. I thought this might be a good section for you to talk about your contribution. If it's okay, we could hear from John and Brad and Dave about the conversation we were just having, but Ken, let’s just hear from you about your thoughts.

Versprille: Sure, that's fine with me. Very good.

Brock: Great, thanks. Okay. Jon, you want to get in.

From 2D to 3D: 1980s

Jon Hirschtick: It's amazing to me that we're talking about the 1980s and we've gone this long and haven't mentioned the two “ginormously” significant things that happened in the 1980s in CAD that are enormous to this and must be a featured part of the history. We all know what AutoCAD is, and others can speak about it much better than I. I remembered it appearing on the market when AutoCAD changed the CAD world in profound ways. Not taking anything away from everything Peter said, I agree that it is all true, but there were a lot of companies like that, Peter. I know you're experienced with many of those. But AutoCAD did something that's amazing, which is they took CAD to the PC. Carl [Bass] is the representative here, and it would be great to get someone who was at AutoCAD at its introduction, too, into the room but that would be another story. Anyway, huge moment. Others can talk about it. Then you have Pro/ENGINEER. I happen to be working at PGC now. I'm sitting in a PGC office, but even when I was at SolidWorks, I always tipped my cap to Pro/ENGINEER as being an amazing example of a system that finally got 3D modeling to work. You can see 3D solid modeling, but just making 3D modeling generally workable. We can talk about the technology and parametrics and feature-based. I know that I was in the research labs and everything else.

I still remember seeing it and my jaw dropped. It was the first system, and that changed the game. Today, it's very hard to find a CAD user today that won't be using innovations that came from AutoCAD and Pro/ENGINEER. AutoCAD for the platform of PC—and yes, there are many other systems besides AutoCAD—is the dominant one that emerged. Pro/ENGINEER for the modeling paradigm. We can all get into debates about the modeling paradigm, but we all use it and we all improved on it, and that changed the world. The result was CAD moving from maybe hundreds or thousands of users, many of them were just doing experimental work, into hundreds of thousands and then millions of people that really got their job done, which is ultimately what we were all trying to enable. Then it also moved, as Peter correctly pointed out, from a 2D way of thinking to messy 3D stuff that didn't really work, into 3D being a standard tool. That's my view of the 1980s history.

For the 1990s, certainly I would say SolidWorks deserves a mention, too, in the history of things. I showed Brad and Peter SolidWorks, and I said, "Look we took the ProEASE model of modeling technology, AutoCAD's business model, and the new Microsoft Windows platform and we made a system that combined the best attributes, and guess what? The market decided that was a very important event. Again, today, I'll argue that every system uses something, but SolidWorks was not until the 1990s. So, I want to go 1980s first. I'm trying to think of a fourth thing in those two decades that would compare to the weight of those three systems. Again, built on the shoulders of a million other things that happened. Versprille to Ian Bray, to Dave Gothard, to Bob Condon. But those three systems defined the 20 years in a way that is really, really remarkable.

Holtz: First of all, Jon, that is dead on. I saw Pro/E at I think either AutoFact or NCGA in 1989, and it just blew my mind at that point in time.

Payne: It was AutoFact.

Brad Holtz: I want to point out a couple of other things that are parallel to that. In 1982, IBM brings out the PC. November of 1982 is when I first saw and first used AutoCAD. And 1982/1983 begins the death of the MiniCAD-based reprographics side of CAD and the explosion of what is, at that point in time called MicroCAD with Autodesk, DataCAD, CADKEY.

Jon Peddie: And VersaCAD.

Holtz: Yes, VersaCAD. The VersaCAD story is really fascinating in terms of why it took the trajectory it did. So, that's the first big thing there. In my mind, there was the introduction of the Silicon Graphics workstations after that that greatly accelerated things. The next key thing in my mind, not on the software side, but on the software-enabling side, was, and I mentioned this before, 1994 Windows NT-3.1 becomes a stable platform and Intergraph makes the jump to say, "We are going to Windows only from that point forward." That was a significant trigger point.

During this time, in the 1980s, computers were precious and expensive, and people were cheap. When we exit 2000, people are expensive and computers are cheap. And that is a very significant happening. Again, Jon, the explosion of the enabling technology of the MicroCAD community from Autodesk and its peers and competitors is that democratization and the explosion and dissemination of CAD to the masses. So, that's my two cents.

Issues of Scale and Complexity at Boeing

Dave Kasik: Okay, I may steal a little bit of your thunder there, Jon, but there were a number of things that happened, at least from my niche, my perch in Peugeot Sound working for Boeing. First, there was still significant heterogeneity in terms of computing platforms that includes graphics platforms, processing platforms with the workstation explosion that occurred during the 1980s. Despite the introduction of the PC, workstations were big enough to be able to handle problems that a company like Boeing had. The democratization and the cost didn't necessarily apply until starting more in the 1990s. Getting software to work in a device-independent, database-independent, graphics- and hardware-independent manner was a real nightmare, and I lived through that first-hand. It was non-trivial. There was this whole piece about trying to deal with high complexity design as the Boeing company had to do, and the question of scale, as Jon Hirschtick pointed out, is really an interesting one. Mainframes were still highly useful during the 1980s and 1990s because scale kills. Every system that Boeing had tried to use failed in terms of minicomputer or workstation-based products, and there was no way that you could design something as complex as a commercial airplane using standard off-the-shelf stuff. That's evolution number one, problem number one.

Problem number two is the entire notion of how you create surface-bounded solid models based on non-uniform rational B-splines. Boeing also took a swing at this. I worked with a group that was led by a guy named Ed Edwards, and he put together enough math talent, people like Eugene Lee, Dick Fuhr, Dick Rice, and Bob Blomgren, who is still active in the world working with Wayne Tiller now on stuff that NVIDIA just bought. These folks were really important in terms of trying to get complex geometry that you could have to all the problems that we've alluded to, stitching surfaces together in a way that would cause a solid to happen, but they succeeded in doing so in the mid-1980s. That was a really good deal in terms of getting the math right. I was thrilled to hear the comment about database management that SDRC also tried to deal with. It was important to get all this work done in a way that would let it happen, that would let you have the results computed and then reliably saved. File systems just didn't cut it.

The last point that I'd like to make is about the advent of the arts and entertainment world in terms of pushing modeling further than might have happened in the ECAD and MCAD world. What these folks were doing in starting to get 3D into even more commodity use is a really big deal. They had requirements that the CAD companies didn't necessarily get, but they still needed the math in order to expand the marketplace and the numbers of users and viewers that we wound up doing. As you all mentioned there was a profusion of different CAD companies that existed.

The other problem that I had, that we've kind of danced around, is that CAD can't live in isolation. Boeing struggled, as I mentioned yesterday, with trying to get data from Gerber to Computervision and back again. That begat IGES. The same problems exist today. Pro/E, CATIA, Unigraphics, and UGS all have the problems of transferring data between and among systems. And the translator market brings on unique, I'd argue, trans-mathematical problems as well as just getting it right and having the data live for a long period of time. That notion of the data that supports products that are long term is a really hard problem. And I can share some stories about Boeing trying to move from CATIA 3 to CATIA 4 to CATIA 5 to the current instantiation of CATIA that computer companies, vendors, that CAD vendors don't even do right.

CAD Industry Standards

Peddie: I have two points. One is I want to give Jon his fourth point. He was struggling to get a fourth one. That is that Autodesk established a standard file format, DWG, and I think that enabled the explosion of CAD as much as CAD running on a PC in itself did. The thing that Autodesk took a little while to work it out is that they stabilized that file format. They stabilized it so successfully that everybody started copying it. Then Autodesk tried to say, "Oh, no, this is ours." But the toothpaste was out of the tube, and they couldn't bring it back. They didn’t officially open it, but it did establish a standard. We have that standard now, and it enables future growth.

Another point I'll make is to build on what Dave was just saying about the entertainment industry; it is when CAD first became a processor-related operation, whereas before it was basic XYZ geometry. With the advent of entertainment, we added a new component called a normal. A normal gave us surface characteristics that involved reflections and subflections and so forth as well as leading us into retracing. Prior to that, there was a demarcation line between CAD and drafting and drawing and what was called post, where you made pretty pictures out of those models.

With the advent of putting the normal in the upfront geometry transformations, you now had a contiguous pipeline that carried that data all the way through so that the normals were accurately tied to the base model. And that was a major breakthrough and also a major, major pain in the ass because it was really hard to do. In fact, we're still working on it, believe it or not.

Carl Bass: Yes, building on Dave and Jon Peddie, since we gave proper recognition to many of the academics who worked on some of the modeling technology, and the genius foundations there, the other group that I think is really important and we haven't mentioned yet but have alluded to is an entire group that immediately pushed the limits of the computer graphics. It started in the academic world with people like Andy van Dam at Brown University, Don Greenburg at Cornell, Turner Whitted at North Carolina, and Jim Briand and a whole host of people at the University of Utah. In order to make this a valuable tool for users, we needed the underlying math, we needed the hardware power that we've talked about. But what made it compelling for a lot of the users were all of the graphics algorithms from hidden line to hidden surface, as Jon pointed out, all of the shading algorithms. Turner deserves huge credit for retracing. The work in Cornell on radiosity and global illumination. They were partially pushed by people on the design side, but as Dave correctly pointed out, it was the people, probably more so on the entertainment side, who really wanted this realism.

That's where you get this interesting divide in computer graphics, and the thing becomes a split between those who want things to look good and those who want it to reflect reality accurately. There's this divide where people go, "I can make stuff up like shaders that make it look like fire or wind or whatever physical phenomena." And there's another very strong philosophical camp that says, "My job is to mimic reality as closely as possible." That leads to things like radiosity. In the spirit of this history, all of those people and the people who went through those labs and studied there are important contributors to what became commercial products.

Interoperability and Referencing

Payne: I want to follow up on something Dave said.

Brock: Okay, sure. Why don't you go then, Michael?

Payne: Much as DWG works in transferring data in 2D, it isn't a solution in 3D. I definitely understand his frustration with the various revisions of CATIA when they completely changed the product. But the worst part of the interoperability problem of which you speak is not the geometry. I ran a group that made translators, and the translators were pretty good after I fixed them. but you can't 100 percent transfer every piece of 3D geometry. You can do it 99-point-something percent in the vast majority, except for the people with their trials.

The main reason you can't transfer data from one of these products to another is because of how they do referencing. They all do referencing in (a) a different and (b) a proprietary manner. That's their secret sauce that they don't want to tell other people. I see Carl agreeing. We know it's true, Carl. In other words, if you want to transfer data from B5 to Unix, good luck. You can transfer geometry, that's it.

People have called it design intent. That was never what we called it at PTC, but somehow the sales guys came up with that term, which wasn't what we meant. You know, you're designing a fuel tank for a car; one intent would be that it contains that much fuel. That's nowhere in the parametric model, but that's beside the point. It is hard to do the transfer of data in 3D in a meaningful way. IGES and Step and all those other silly things are not a transfer mechanism worth a damn because you can't do much with it afterward.

Peddie: What about Colada?

Brock: Dave, I see that you want to get in here. Can you just take like one minute, literally? And then let's turn to Ken.

Kasik: The comment about the data transfer is that when you're dealing with 200,000 parts, 99.9 percent, which has never been achieved at Boeing, ain't enough.

Payne: That's right!

Kasik: You have to go back and retouch them by hand. It's like human translation, human-language translation. You screw stuff up. The bigger problem, at least that we encountered at Boeing, was getting the product structure correct because different people had different product structures.

The last comment is about the commoditization of GPUs for the games in the PC industry, which made the cost of hardware so much lower, astronomically lower over the period that we're talking about. That's what really allowed PCs to be used in the CAD business, and the CAD business rode on the coattails of the games business. Let's be real. They made it. They made GPUs cheap. Period. A minute. How's that, Dave?

Brock: Perfect. Perfectly done. You nailed it.

<u>Develoment and Adoption of NURBS</u>

Brock: Ken, if you'd like to take the floor. Just tell us what you'd like us to know about your journey in the world of CAD.

Versprille: In CAD, what I'm obviously most known for is NURBS. Although I think I did quite a bit of contribution during my years at Computervision in the R&D organization. The story of NURBS is actually written down in a number of places, but a quick synopsis is I graduated in the late 1960s from University of New Hampshire as a math guy. I had just been in that final year exposed to computers. It was a big honking IBM machine. I was intrigued by it, so I decided to go to graduate school and focus on computers and the math. I didn't even know CAD existed at that point in time. And what happened was what I call a lot of things fake happenings. I actually had to go pay my own way for grad school, so when I applied, I applied for money.

I got an NSF [National Science Foundation] scholarship at Syracuse University. I was from upstate New York, so I decided to go there. They assigned graduate student professors for graduate students as their advisor, and I lucked out by pulling Steven A. Coons, who is now considered the Grandfather of Computer Graphics. He had come from MIT. I took his course, and being fairly savvy, I said, "Okay, he's my advisor. I should probably take his class." By the third class, I knew exactly what I wanted to do for the rest of my life. It was basically Computer Graphics, Introduction to CAD.

At the time, it was sort of the golden age of surfacing. At that time, he brought in Bill Gordon from General Motors, the Gordon surface. He brought in some professors from Cambridge, England, and there was a high intensity amongst a very small group of graduate students working on surfacing. Bezier's math had just hit the industry, but it didn't seem to be taking off all that much, at least in the beginning, in the first few years.

One of the grad students, Rich Riesenfeld, who was like a year or so ahead of me working with Bill Gordon, came up with a spline version of Bezier, which they called B-splines. Rich got his PhD based on that and went off to the University of Utah to teach. I followed up by taking the next step from the B-spline, which was to go rational and go nonuniform in both directions. Most people were designing with what was called polyconics at the time for curves and surfaces, which was stringing together with tangency conic sections. I focused in on the rational case to try to show that you can transition from conics over to NURBS and keep everything exactly the same. It was a whole new methodology for working with the math. Got my PhD based on it. Went off to Computervision because Steve Coons had been an advisor to them for a number of years. Got into the R&D organization, and within two years, I had surpassed my boss and really took off there. I was loving it.

Suddenly, about 10 years later, I started getting checks because copies of my dissertation was being sold by the University of Michigan. I said, "What is going on?" It turned out Rich Riesenfeld had trained or taught B-splines and NURBS at Utah, and a couple of their students went on to Boeing.

Boeing at the time had a problem: they had a rich set of math applications, but most of those people had died or moved on, and they were having trouble moving data between the applications. They jumped on the fact that NURBS could preserve the shape of the geometry between different applications that they had been building. I was told the story that Boeing then demanded that their vendors implement NURBS, which is when it took off. Everybody jumped on that bandwagon.

I was actually very surprised because my dissertation was published in 1975 and it was much too complex for real advanced use. The rational was good, but the nonuniform would not be necessarily of use, at least back then. Of course, things have changed. It's been 40 years now, so what can I say? That's really the story.

Brock: I know that Dave and Peter want to get in here with questions. I think I'd like to just jump the queue and ask you if you could just speak for a moment or two about what you think it was about this approach of NURBS that accounted for its attraction to people as for translating between these different programs and platforms? Explain to a lay audience what it was about the mathematics that you think gave it this utility that it would come to have. Is that a fair question?

Versprille: Like I said, at the time any of the advanced curve surfacing stuff that was done was what they called polyconic. It was composed of conic sections. The big thing in my dissertation with NURBS was I proved mathematically that a NURB could represent a conic section exactly. Absolutely no approximation. Exactly. That's what interested a lot of the users who were doing the polyconic stuff.

It was a new way of interacting with them, which is what I credit Bezier with doing, which was the polygon or whatever you want to call it. In Europe they call it something different than what they do in the States, but B-spline and NURBS maintain that kind of capability working with a polygon for a curve and a net of polygons for a surface. It slowly got adopted with that whole new user interface thing. At the root of it, it was the polyconic thing. Once people got into it, I think they started to look at curvature so they went to cubic and beyond.

Kasik: Right. As the token Boeing guy here, Ken, it was my group that was working on the NURBS stuff in the late 1970s. To clear up a couple things about what you said about Boeing. Boeing did its lofting job using conic sections. It was a program called TX-95. People were retiring, but the problem wasn't necessarily transferring from TX-95 to other places, it was trying to get a good representation so that Boeing could start doing more design than drafting as it was doing in the late 1970s. The group that I led developed the fundamentals of proving that nonuniform rational B-splines could represent the conic sections precisely. Exactly. That's what caused the landslide of money to you.

Versprille: I made like 30 bucks.

Kasik: Hey, that was more than I made off the books that I've written. But the point is that the math group that worked for me had some really good folks that were from Utah. Jeff Lane was one of Riesenfeld's students.

Versprille: Yes, I was thinking of Jeff Lane. That name sticks in my mind. I was told Jeff Lane was the impetus who took the NURBS idea.

Kasik: Yes, Jeff brought NURBS from Riesenfeld to Boeing. He was also responsible for rendering bicubic patches with Loren Carpenter as well during that time. Loren Carpenter developed fractals while he was working for me as well. It was a really interesting group of folks. I'm not sure you heard, but Bob Lundgren, Dick Rice, Eugene Lee, and a bunch of really good NURBS folks worked for Boeing during that time.

Versprille: Yes.

Kasik: Thank you for introducing non-uniforms. But it really was not a transfer problem. It was being able to say that Boeing wanted a CAD system that would allow people to represent conics correctly because that's the way that ships for water travel and airships had been designed since the early 1900s. And it all worked, so that was just a little clarity. Thank you

<u>Downfall of Computervision</u>

Marks: Ken, the question I had for you is, given that NURBS, as Mike would say, was "only temporarily," but it more or less became the de facto standard for the underlying representation, and given that you were at CV, can you tell us a little bit about why CV isn't the world leader today? What was their adoption? We have all these cases of the gem sitting in the company, and they're looking elsewhere for profits. What kept CV from jumping on this with both feet?

Versprille: I think it's a very complicated story as to what caused the downfall of CV. I think they missed the boat in a number of places. You have to remember at that time, in the 1980s, they were selling both hardware and software. A big group, Hardware Engineering, was created with a vice president who was one of the top vice presidents at the time. Both he and manufacturing fought the move toward Unix and a workstation. They fought it for two years. Finally, they had to give in, but it was already too late. All the competitors had moved to the workstation.

Peddie: Was that Ledeen?

Versprille: Yes.

Holtz: There's another aspect here: Computervision was entrenched in the Navy and in 1988, they put out a bid for a new CAD system that they were going to use for the next 10 years. That was the CADDS 2 bid. It was a multibillion-dollar bid. I was engaged for a good bit of that actually. Consulting hours on that built a good portion of my house. I was on the team working with Ken Ledeen and with a company called ATI—a different ATI—that did that with Computervision. We got 994/996 points out of 1,000 possible on that, but we lost the bid to Intergraph. Afterward, I learned through the grapevine that the Navy was so frustrated with Computervision that there was no way they were going to let Computervision win the award.

Subsequent to that, 1993 I think, there was another government bid that Autodesk and Intergraph bid on—Carl, I think you were there at the time. That led to an award to Intergraph, and Autodesk protested. The decision was the first implementation of the term called “coopetition”: "All of you can come into this, but you've got to be able to work together and the systems need to interact." Carl, do you want to elaborate? You remember that?

Bass: I do remember. Yes, I think you got it right.

Holtz: This whole idea of interoperability became the primary focus of everybody's attention, and it helped us launch COFEs, the summit of the future of engineering software in 1989. The whole purpose of that was a discussion of interoperability.

IEEE Annals Special Issue

Marks: What are you planning to do in terms of this history?

Grad: We have two things. One, it will all be transcribed and then posted on the CHM website. And, if you guys are interested, we could do a special issue of the *IEEE Annals*. It'd be about the 11th one that we would have done. If some of you here are willing to write articles describing your experiences, it can get published probably in about a year or so. It would take some work on your part, but I'd be glad to coordinate it with the historian here, Daniel Llach. if some of you are willing to volunteer to work with us. It's been a very exciting thing to put these whole issues out devoted to one software subject.

Peddie: When can we get the transcripts?

Grad: I'll have to ask Massimo [Petrozzi] that, but usually the transcripts get done in about two to four weeks. Then we edit them to clean up your dirty words and things like that.

Payne: I appreciate that.

Peddie: We need the transcript sso we can stay consistent with what we said, because in one place we might say we invented the sun and in another place we might say we invented the moon.

Grad: I'm not going to change that. If you made a false claim, you have to live with that for eternity.

Marks: I love the t-shirt that says, "Let's make lying wrong again."

Grad: If you some of you are willing to do it, I'll solicit that after the meeting, because this is great content and all of you have so much more to add. The articles don't have to get peer reviewed, but they are reviewed by people in the industry for accuracy.

Peddie: Different kind of peers. But you have amongst this group some really outstanding writers. I won't embarrass them by saying their names, but they really are good writers, and if you can get them to sign up, you're going to have a nice piece of work.

Grad: I think that Jon, you and Joel and Brad could also contribute considerably. You guys do know how to write.

Peddie: Well, so does Mr. Marks.

Holtz: Well, Joel does, in any case.

Marks: Dave is well prepared to do this. Jon is actually an outstanding writer. You talked about the things, but he has a secret weapon and that's his wife [Kathleen Maher] who's possibly even better.

Grad: I'd love to do it, but I wouldn’tt said "Yes" until I could hear what you guys had to say here.

Peddie: You kidding? This group of sweat and egos? Of course, we'll do it.

CAD Company Leadership

Marks: I don't know how you tell the story, and maybe it's not appropriate for your venue, but it is really hard to have a CAD or any software company that lasts forever. As Jon Hirschtick was saying, they're not supposed to last forever. But there's this general trajectory for a company: invent some really cool stuff; buy a bunch of Ken's theses; decide to go to market; and then get kind of screwed up when they try to financialize it and put it out of business. Jon Hirschtick, they did some excellent technical stuff, but they just did an amazing job on the customer front, and Carl didn't have as much freedom to run the company the way he wanted, but in my mind, he was the thing that kept Autodesk from turning the edge over, losing sight of its customers and the rest of that. I hope Carl you're going to talk about this. In terms of all their CEOs, Carl struck me as the guy who balanced, "Let's do something right for customers. Let's keep the technology going. But let’s keep the doors open by making a profit."

Grad: If I may add to what you're saying, Peter. We have done four oral histories of the Autodesk CEOs. We didn't do John Walker, because he's in Switzerland, but we've done four of the founders, Al Green, Carol Bartz, and Carl [Bass]. It's a wonderful story. We'd love to do oral histories of some of you guys, too. That's up to David Brock, but they're good stories to tell. All of you guys have great stories. So many software companies screwed themselves up. They did one thing right, and therefore they thought knew what the hell they were doing.

Peddie: I give Carl credit at Autodesk for managing impulse control.

Payne: In a way there's no difference between the evolution of the CAD companies and Sears. It just took much longer for Sears to disappear, but they were going to disappear because they didn't move with the times. They didn't recognize the changes, that the next day wasn't the same as the previous day.

Marks: Michael, who in your history is K-Mart?

Holtz: But there is something to be said that old CAD systems never die.

Peddie: Is it still that way?

Holtz: It's still that way. CADKEY, DataCAD, and Evolution Computing; they are still selling copies because their users are still using it if they haven't transitioned.

Solo and Small Groups of Developers

Payne: One of the things that we missed, and marketing people are very good at missing things, is that there's a continuum of complexity in mechanical design that ranges from making simple things to making airplanes, if you want me to take that as the ultimate assembly. There's a lot of bits and pieces in a 787, for example. But a product that can just do simple things could last for a long time. But when a product can't do the high end, it tends to fall out of favor.

Kasik: It's true. One of the interesting things I meant about Carl's comments and other comments about Boeing forcing NURBS on the world, basically showed the clout of a very large company, some of them actually hire some people who know what they're doing to try to make something better. Because left to its own devices, the marketplace for a commercial companies would not necessarily have adopted different forms of mathematics or other things without the push from the customer base.

Marks: Yes. I've got another question in terms of the overall flow. We have these multiple parallel tracks we talked about; government, companies in technology, and universities, and so on. But we haven't talked about hero solo programmers. You got guys like Mike Riddle early in the history of Autodesk. Would Sam Geisberg be one of these guys, Michael?

Payne: There's another one I can add to that list: David Taylor made a CAD system all on his own and sold it to Computervision.

Marks: And VersaCAD. Who was the guy who did the first variation?

Holtz: It was actually Tom Lazear's son.

Peddie: Tom and Michael.

Holtz: Yes.

Marks: Who was the guy who did the first variational 2D out of the UK? I'm having a brain skip here.

Grad: Bentley?

Holtz: Yes, Keith Bentley.

Marks: No. It was somebody else, somebody else.

Bass: Peter, the only thing I'd say is most of these things were done with groups of people, but history rewrites it as these individuals.

Marks: Yes, good point.

Bass: Even when you go back to the university labs and there's a name on the paper, it's the five grad students they had coffee with every day who contributed.

Peddie: Yes, the first name on the paper is the damn professor who didn't even do it.

Bass: Right, exactly. I think there's a lot of mythology about the solo programmer. Particularly to make commercial systems, it's like someone could have the idea like Sam, but how many people worked on it for like 20 years until it really did what was needed?

Brock: It’s time for a short break.

END OF DAY 2 SESSION 5

