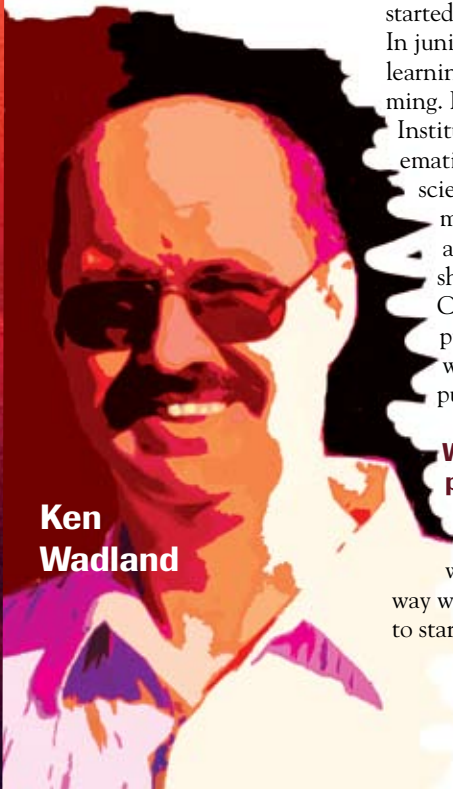




In 1995, OrCAD had the most popular PC-based PCB (printed-circuit board)-schematic tool. Another popular tool was available from PADS, which Mentor Graphics now owns. Engineers designed PCB schematics in OrCAD and laid out the board in PADS, a clumsy approach that still persists to this day. Rather than redesign its PCB-layout tool, OrCAD purchased Massteck, a small company offering a good layout tool and, more important, an autorouter that achieved 100% routing on complex boards. PCB designers Al Akermann and Gene Dancause, Massteck's founders, contracted with Ken Wadland, a computer-science guru, to do the autorouting algorithm and database architecture. Just as vital, they hired board designer Chuck Grant, who first did product testing and then eventually wrote software. Cadence in 1999 bought OrCAD. Sadly, OrCAD no longer supports the original Massteck-based OrCAD Layout tool, instead favoring OrCAD Editor, a stripped-down version of Allegro, Cadence's enterprise-class layout software. Wadland and Grant now work for Cadence in a Massachusetts design center. EDN recently had the chance to interview them.

Improving on PCB design

Ken Wadland and Chuck Grant: the innovators behind OrCAD Layout software



Ken Wadland

Where did you get your education?

CHUCK GRANT: I got an economics degree from the University of Rochester. Other than using some computers in those courses, I hadn't been interested in them. But the PCB-design business was growing by leaps and bounds, so they were looking for PCB designers. I just went in and I liked what I saw.

KEN WADLAND: In seventh grade, I started Wadland TV and Radio Repair. In junior high and high school, I started learning assembly-language programming. I went to Worcester Polytechnic Institute and got my degree in mathematics, but I took every computer-science course they offered. I got my master's degree and my doctorate at the University of New Hampshire. I went to Pittsburgh State College, and I created the [computer-science] curriculum there. I was the first chairman of the computer-science department.

What gave you guys the entrepreneurial spirit?

CG: There was a lot of evangelism in that we had a unique way of designing PCBs. The way we divided and conquered was to start from the most difficult part of

the design and completely finish that area. Once you clear out that most difficult area, you just move on to the next one, and you keep stringing them out until you get to the end. When you get to a clear area, you could pop in a via and use the other layer to finish it. We knew that the value added for PCB design is in autorouting. So we said, "Let's take the technique we use ... to divide and conquer as a manual designer and apply it to software."

KW: I couldn't resist the allure of money. [Autorouting] was the only reason I was attracted. At that point, there were only two hard problems in the world: AI [artificial intelligence] and autorouting. Autorouting is a mathematically complicated problem. Beyond being mathematically complicated, there are the aesthetics and the electrical-design rules and everything else that made it just a challenging product.

What made your innovations so effective?

CG: In a start-up company, you wear a lot of different hats. The hats I wore went from testing to marketing to training, sales, and customer support. One of the nice things about a small company is that you really can try all those

things out and see what you like and gain a lot of experience. We had some good PCB designers in that company who were proficient, as well. I decided that I would like to learn how to be a programmer, and, of course, I learned from the best, which is Ken, so that was a great opportunity for me. [Innovation] was driven by the PCB designers because we were used to the time pressures and deadlines and the need to get it right the first time. [We felt] “Hey, we get to invent our own, so let’s do what we want to do, and this is what it’s going to look like.”

KW: Well, we did not have a marketing group separate from our PCB designers. All the people in sales and marketing were PCB designers. So when they gave us the requirements, they understood what they were talking about.

What drove your product innovation?

CG: We really had to be evangelists in interactivity. The marketing machines of all the CAD companies were touting “push the button.” That was their idea, “push the button, and we’ll route your board for you.” Well, real PCB designers know that [approach] never works. You get as much automation as you can, and if you can guide it, all the better. The interactivity we built in was that you could choose where on the board you wanted to start routing. You could start in two different places and have them join up in the middle. So there was interactivity, and then there was the push and shove, which everybody copied. The idea that you could pick a single line, route it between two other lines, click, and [have] everything push out of the way was something that no one else had. We would let you see what it accomplished, and, if it didn’t finish, you might say, “Well, I can fix up that mistake over there, or I’ll take care of it.” But we wouldn’t break any design rules. We would leave unroutes where we couldn’t get them to fit. If you didn’t like it, you could go back to what you had before. We had an undo right from the beginning.

How do you become an innovative engineer?

KW: I don’t follow rules. I don’t like to

start a project unless at least one other person failed at it. Why waste your time on something easy? The downside is: You fail a lot. To do innovation right, you have to ignore the details until you have already established the concept, and then you see if the details work. If you’re too detail-oriented, you reject it too soon. You think of really outlandish things, most of which won’t work. You get a bunch of outlandish things, and then you start pruning them down to one that can actually be usable. If you can’t suspend disbelief for a long enough time, you can’t get these other ideas. One of the things I teach in my classes is: Never write code until you’ve found three ways of doing it. A lot of people [take] the first [approach] they think of. Chances are, it’s the wrong way or a less-than-perfect way. So you want to find three approaches to the problem, analyze all three of them, and then pick the best. People just don’t do that. They just don’t take the extra time to do that extra step.

Why was the program so intuitive and easy to use?

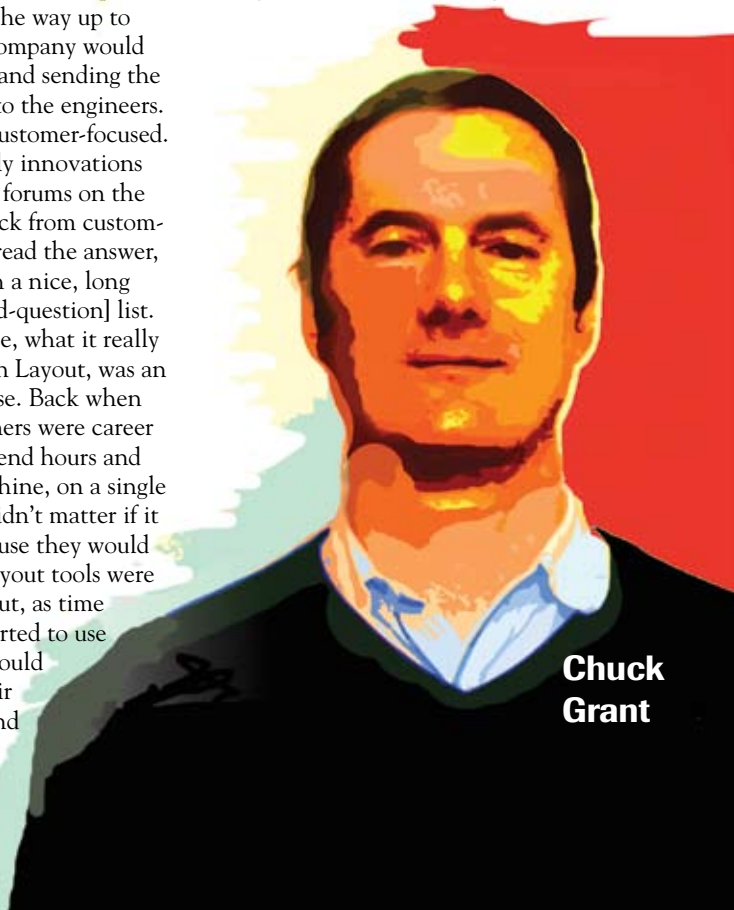
CG: [It was not] that some marketing spec had been translated through an engineer. It was a very direct experience. Those of us all the way up to the president of the company would be using the software and sending the feedback to Ken and to the engineers. We were very much customer-focused. I think one of the early innovations was to use interactive forums on the Internet to get feedback from customers. Everybody could read the answer, and you’d end up with a nice, long FAQ [frequently asked-question] list. The OrCAD influence, what it really brought to the table in Layout, was an emphasis on ease of use. Back when we began, PCB designers were career people, who would spend hours and hours on a single machine, on a single piece of software. It didn’t matter if it was complicated because they would learn it eventually. Layout tools were plenty complicated, but, as time went by, engineers started to use the software. So we would take into account their wishes and requests and tried to make it easier

to get into, and a lot of that happened after the purchase by OrCAD.

Users love the spreadsheet functions in OrCAD Layout. How did you come up with incorporating spreadsheets in the product?

CG: We were unique in that we leveraged existing technology before a lot of other people ever thought of doing that. We knew about Lotus 1-2-3, which we used on a PC for our own accounting, so we leveraged the idea of spreadsheets, not really realizing that they would take over the world later. It was just something that looked like it would work well for us. It allowed us a lot of flexibility. We could lock and unlock nets, establish library components or library symbols, and attach them to the components and all the things we needed to do. It all worked well with that spreadsheet interface, so we went to town with it and used it because it worked so well.

KW: Making a system that’s simple is very hard. One of the early winners was the Lotus 1-2-3 spreadsheet. There was some brainstorming between Gene Dancause and me, but it was my attempt to write a minimum number of lines of software to get maximum capability. We wanted the ability to make



Chuck Grant

“Every time we got a board to 100%, we jumped up and down. Getting 99.97% was cool, but if you hit 100%? Wow, we were excited.”

changes to more than one object at once. So we came up with this spreadsheet because our database resembled SQL [Structured Query Language]. It was kind of a relational database under the hood, so producing a spreadsheet was a piece of cake. And then editing it through the spreadsheet was just a lot less software to write than writing dialogues for everything you wanted to do. It was basically being lazy.

How did you conceptualize the color-rule interface?

CG: The idea was that the program was WYSIWYG [what you see is what you get], so you could say, “OK, I want to turn on these objects.” This might be the etch and the pads on the top layer, and that’s my Gerber [file], and then you just print it out. So then you have a set of these [files], which are on a spreadsheet, and each one of them establishes a set of rules [of] what’s going to be visible for that artwork layer. If you have the assembly layer, the silk screen, and all of those things that also come with a PCB, ... that’s what will print out on your Gerber.

KW: The problem they posed to me is: “We want to see the colors in different ways at different times.” Some engineers wanted to see the top layer red; some wanted it green; and some wanted to see this [color] differ some of the time. I told them “I can’t do all that. How about if I just give you a color set and you can change the color set when you want?” The other thing that came into it is my campaign to protect people who are colorblind. My son is colorblind. So I refused to have any products that had fixed colors. Everything had to be adjustable. We came up with this idea. We call it “color set.” You set up all the colors the way you like, you save it, and then you pick from those color sets.

Was there any innovation due to the limitations of a 286 PC, and did you program in C?

CG: It was all C. The other innovation that we had was in the way we did software best practices. This [idea] was Ken’s innovation. We ran C, but we

isolated the database from the interactivity by using an object-oriented type of programming so that we prevented database corruption, which was a common problem. In most CAD systems, the programming was done so that you could manipulate an object directly. [We made] a copy of the object, let the user manipulate it, and then let the user commit that [copy] to the database. It would all happen under the hood. What that allowed us to do is to recover from errors without the user’s being aware of it.

KW: We knew from the beginning that a 64-[kbit] database was not going to be big enough. Back then, we had expanded memory and extended memory. One of them came in banks, and the other one worked only in 32-bit mode. We took this “get-put” methodology: ... Under the hood, when you say “get,” we’ll switch in the right bank, or we’ll switch to 32-bit mode and get the data for you. When you say “put,” we’ll do the same game back again. So the application lived in the 16-bit Windows environment, but it could access unlimited memory. We had a complete separation between the UI [user interface], the algorithms, and the database. I had used that concept when I worked with relational databases, so I understood the concept, and I applied that to an object-oriented database.

What was the biggest problem you felt you had to take on?

CG: The biggest problem in marketing was that the PCB designers don’t make buying decisions. They can influence them to some extent, but upper management and engineering make most of the decisions. It was difficult to convince people who’d never done PCB design that interactivity was critical because someone else would come in and say, “We don’t have interactivity; you just push the button.” Well, you can push the button on ours, too, but that’s not really what you want to do. I think we did eventually win that war, where interactivity became important to people.

KW: I’ll tell you the biggest problem that didn’t get solved. We tried to get a good place algorithm. We tried to have

an autoplacement algorithm, and we put enormous resources into it. It’s a simpler problem mathematically in the IC world, where everything’s symmetric. It’s a problem in the PCB world, where every component is a different shape. Things are not interchangeable. If one’s a quad pack, you can’t switch that with a two-pin surface mount. They’re incompatible. So we failed in placing. As far as I know, no one has ever had a successful placement tool in the PCB field.

Was there a personal victory you had?

CG: I will never forget one of those sessions when I had one important customer who absolutely needed to have the ability to move components and have minimization take place as they were moving. [We had] the first product I know of that had interactive component moving and then [PCB]-trace minimization as you were moving it. Initially, it was just a rat’s nest, and then we had a customer who said, “I have to see the real traces.” I definitely remember that one.

Was there a company victory you remember?

CG: In 1994, Jeff Hawes, [a former application engineer at Cadence, OrCAD, and Massteck], and I went to the PCB Design Conference and we entered MAX EDA into the competition. They were assigning points to PCB-design systems, and what they did was pretty comprehensive. You had to design a board, sight unseen. They gave you a net list and the parts. You had to design the board, create the netlist, bring it in, autoroute it, and print out Gerber artwork—all within two days. ... We won that contest in 1994 and 1995, and that was the year OrCAD bought us. When we found out that we had won that competition, we were elated.

KW: I’m not sure there’s any one [victory], but basically every time we got a board to 100%, we jumped up and down. Getting 99.97% was cool, but if you hit 100%? Wow, we were excited.

—interview conducted and edited by Paul Rako