Hank Korth Speaks Out on two-career issues, why not to write a book in the beginning of your career, and more

by Marianne Winslett and Vanessa Braganholo



Hank Korth
http://www.cse.lehigh.edu/~korth

Welcome to ACM SIGMOD Record Series of Interviews with Distinguished Members of the Database Community. I'm Marianne Winslett, and today we are in Providence, site of the 2009 SIGMOD/PODS conference. I have here with me Hank Korth, who is the Wieseman Professor and Chair of the Department of Computer Science and Engineering at Lehigh University[1]*. Before joining Lehigh, Hank was the Director of Database Principles Research at Bell Labs, a vice president of Panasonic Technologies, a professor at the University of Texas at Austin, and a research staff member at IBM TJ Watson. Hank's research interests lie in database systems, especially transaction processing. Hank is on the editorial board of ACM TODS. He is an ACM Fellow and an IEEE Fellow. His PhD is from Princeton. So Hank, welcome!*

[1] Hank Korth served as Department chair in Lehigh from 2003 to 2009.

During your time at Bell Labs, the company underwent a lot of surgery. What was it like working there?

I started at Bell Labs about a month before AT&T had, what they called, trivestiture. This is where AT&T split themselves up into a new AT&T, not the current one, but a new one then, what became Lucent, and NCR. So, there I was, a new employee at Bell Labs, wondering what is this all about. "Is this an excuse to get rid of research? What's going to happen?" It took a while to work it out. As we all know now, AT&T had its own research lab, AT&T Labs, which is still there, alive and well. Lucent inherited the Bell Labs name that still continues, although today in a much different form. But then came the telecom boom, when Bell Labs was a truly wonderful place to be in research.

So, you were there during that golden period?

Exactly. It turned out to be the perfect time.

At Bell Labs, you were involved with technology transfer. It is hard for many big companies to get their research results out into products. So, how did tech transfer work at Bell Labs?

Historically, tech transfer at Bell Labs was not very successful. For one thing, in the monopoly days, it was less necessary. Lucent being a profit making company, it became more important. But I really did not get involved in tech transfer initially. We were initially involved in building a new research center under Avi Silberschatz. I built, with Avi's substantial help, a very nice database group. Then, a bit later on, Lucent, flushed with the telecom boom, bought at telecommunication billing company, called Kenan Systems.

The founder of that company, who then came into Lucent to head the software products group to be affiliated with Bell Labs, designed a special organization whose goal was to take mature research projects in Bell Labs, and put a wrapper around them consisting of a development team as well as a business team. A separate organization housed this until first customer introduction. Then the whole package, not just the technology, but now a product, a marketing team, customers, the whole works, would be transferred into the business. I was on the leadership team of that organization, which put me in a very interesting spot, because I was a department head at Bell Labs, plus involved in this. We got a number of very nice ideas productized, that started off wonderfully, but of course, then the telecommunication bust hit us, and we never really got to bring that whole model completely to fruition. But it was a great time, and I think a very good model for tech transfer, because what you are able to do then is isolate the group that is trying to make the technology something real; treat it as something special in the company, so you can attract the most dynamic and interesting people to the group. Then for the true tech transfer, you now have the full organization in place, and people who appreciate the value of the technology, but also appreciate the value of the business end of the technology. I think where most companies fail in tech transfer is that you get technologists who think marketing is trivial, and you get marketers with a superficial understanding of the technology. You wind up with a horrible mismatch, and people unhappy on both sides.

Your database textbook with Avi Silberschatz and Sudarshan, Database System Concepts, is about to have its sixth edition, which to me sounds enormous!

It is. If I think about all the words that we've written over the years, I wonder if I would have started. But, all kidding aside there, the initial idea, way back in the early '80s, was that there was a gap between very practical database books and very theoretical ones. Avi Silberschatz and I (Sudarshan wasn't in the picture at that point) were looking to fit somewhere in between. I did that as an assistant professor, something that I certainly would not recommend to new faculty getting started. But I did that because I had a unique opportunity. Silberschatz had a successful operating systems book, still successful, by the way, *Operating System Concepts*. He wanted to do a database book, and asked if I was interested. I did not think I would necessarily get the opportunity again to co-author with an experienced and successful person like Avi, and he is just a great person to work with. We had already started working on some research together. And so I decided "okay, if I don't get tenure, I don't get tenure, but here's this opportunity, I'm going to take it". So despite the book, I did manage to get tenure, but I really have to emphasize that's not the best strategy for that. Books do not count in the computer science tenure process. Not in any place that I know of.

What about your time in Panasonic? Panasonic isn't a big name in the database community. What made you go there?

Well, I was at the University of Texas at the time. I had tenure, my wife finished her PhD, so we had a two-career issue to solve. Obviously, there were lots of possibilities, but the intersection of possibilities in computer science, and possibilities in the pharmaceutical industry, aren't exactly that large. The New Jersey and Philadelphia area is wonderful for the pharmaceutical business, and it turned out that there was this great opportunity. Well, Panasonic wasn't exactly a household name in computing, but this was a time, in the early '90's, where the Japanese economy had been booming, and the major Japanese companies were looking to have a research presence in the United States. And so, some visionaries within Panasonic had the idea of setting up an information technology lab. It was located in Princeton, it was headed by Dick Lipton, and among the people involved were some people from the database community: Hector Garcia-Molina, Rafael Alonso, in particular; and from the OS community, Kai Li. It was a really wonderful group. Then we added a number of others in databases – there was Daniel Barbará, as well as myself.

Some years from now […] there will be nothing on paper, and that can change the way we do reviews, the way we evaluate publications, and the way we evaluate people for promotion.

This looked like an opportunity to combine the kinds of things we were doing in database research with consumer electronics. One of the things that we paid a lot of attention to was the concept of mobile computing, which was very new then! In fact, some of the things that we

talked about were clearly way too early. I remember Rafael Alonso talking about having the web in the palm of your hand. Okay, but why would anybody want the web in the palm of your hand? Well, it is now a joke, but that we were thinking about those issues back then, and this would have been in the range of 1991-94, made it a truly wonderful time. Wonderful until the Japanese economy went down, and the economics of having these labs began to make less sense for Panasonic. So the lab got redirected towards a more year term applied set of objectives, which then led to many of us moving to other places -- in my case, it was Bell Labs.

With Francois Bancilhon and Won Kim, you wrote a 1985 VLDB paper called "A Model of CAD Transactions". That paper won a 1995 VLDB award as the most influential paper from the proceedings of ten years ago. What was that paper about?

At the time we were working on this in 1985, there was a lot of concern about how to manage databases that contained not traditional data processing data, but design data. The issue there is that, if you are working on a design, you don't simply read a bank balance, add an amount, and put it back into the database. Instead, you read some part of the design, and work on it for some period of time, which could be a few minutes, but also could be a few days. Then you check it back into the database. Transactions in that realm are long duration transactions. And so, we need to have different ideas of how to manage concurrency, and recovery, and define what a transaction will be in that process. So, we were thinking about issues clearly related to long-duration nested transactions, and other ideas along those lines. They were all happening at the same time. Those were the ideas we talked about in that paper.

You've been the department chair at Lehigh for a long time. What have been your goals there?

I came to Lehigh as Bell Labs was becoming a less interesting place to be than it had been, because of the changes going on with the telecom bust, which was followed by the dot com bust, and it looked like a good time to move back to academia. Lehigh was an interesting challenge. The computer science department there (technically Computer Science and Engineering) was formed in 2001. So, I was coming in, basically, at the beginning, and while there were some faculty members there, it was a time to try to mold the department into something that could be a good research department, while maintaining Lehigh's traditional strong focus on undergraduate education. I did quite a bit of hiring, which seems in some ways to have been the story of my life. I did that for the database group at Bell Labs. Then at Lehigh I had to look much more broadly than databases, to have a department with sufficient breadth to be able to cover computer science from a curricular point of view, but also to get groups of people with sufficiently similar research interests, that we have a strong research activity.

I am now done being chair. Today is my second day as not being chair anymore, but in the six and a half years I hired more than half the faculty in the department, and I am really pleased with the results. We have four NSF CAREER award winners in the department, and a department with 16 tenure/tenure track faculty. I think that is quite good. Also, I am very pleased with my first hire, Dan Lopresti, who had been at Panasonic with me, he was at Bell Labs, and he's going to be my successor now as Chair. I know he is going to do a wonderful job. So here the challenges were, reaching broadly, not just in areas of computer science that I know well, but in areas that I perhaps don't know as well as I should. But still, I needed to try to lead the

department in bringing the right people together, and also to try to maintain the sense of community that I found when I was there. That was one of the things that attracted me to Lehigh. It was a wonderful group of people. There was none of the factions that you see in so many departments, but rather a real strong sense that we all want to create a good department. And we are still that way. It has been very happy place to be.

You said that the focus is on undergraduate education. Did you have time to do research?

Anybody who is department chair, you know, there's advice that you get that of teaching, research, and administration – one of them has to go. If the department chair lets administration go, that can be problematic. Because I was coming from industry, I thought it was very important that I do some teaching, both on the graduate and undergraduate level, so that I had a sense of what the Lehigh community was about. We were looking to strengthen our graduate programs, strengthen our research, but I also wanted to understand the Lehigh undergraduate traditions. One of the things that I ended up teaching was intro, which I'll admit, was a tremendous amount of work, but I have a lot of fun doing it. I do get a kick out of students taking their interactive webpage and sending it home to mom by email, and just feeling so proud of themselves, that they've done that.

If you enjoy what you are doing, you are going to be much more productive, and much more

So, that is how you introduce computer science, through webpages, and Java Script, or something?

Well, that's the beginning. The main focus of the course is sufficient Java programming to get onto the second course, as we do our undergraduate curriculum initially in Java, and then switch to C and C++. But, part of the course is to talk about the breadth of computer science that is really a whole lot more than programming. It is much more about design, creativity, and interactivity. All the students have web experience coming in, maybe not web development, but certainly web usage. And when they can actually create something that they can click on, and their browser opens up and runs their code, they love it.

Are there textbooks that take this Intro to CS via let's start at the web?

There are a number of books that do that, some better than others. But we also, of course, need a Java book. So, I pick one of these general books, and then a Java book that we continue to use for the second course. We use both of those, but I tend to do things more my way than any book's way, in part because my audience is different than the typical intro course.

How are they different?

Our engineering students have to take the engineering curriculum, and don't take my course. So I have our arts and science students, and students who are computer science and business majors.

The Computer Science and Business program is an accredited CS degree, and business degree altogether in one. The only major like that anywhere. I have a lot of students that have some degree of business interest, and I have some experience with that (we've talked about my work with technology transfer). So, I try to bring some of that perspective into the class as well. But that's in no textbook.

The VLDB conference recently backed away from double-blind reviewing, while ACM Transactions on Database Systems (TODS) recently took it up. What do you think about double-blind reviewing?

It's far from perfect, and it is very easy to complain about it. But I think the fundamental thing that drove the discussion in TODS is that there are real problems in terms of perceptions of unfairness, and maybe some actual unfairness in the process that we were using, where everything was totally open to the reviewers. Even if it is just the perception, I think it is important to try to be as fair and as open as possible in the process. I think TODS has put together a very nice compromise there, because it is not absolute blind reviewing – when some unblinding becomes necessary in the process, the associate editors have some discretion to try to do the right thing. So I think it works well there. In a conference setting, I think it is more problematic, because there is a deadline by which the decisions have to be reached, and that creates a little bit more pressure in the review process.

Well, you guys have a deadline too, because you do this fast turn-around, faster than conferences, in fact.

True, but the deadline we are talking about there is a 5 month deadline from submission to decision. Conferences have a long turn-around time in part because there is the whole delay between the decisions on the paper, and when the conference actually happens. So, the reviewers have the paper in hand for a longer time in the TODS process than they do in a conference process. And that gives us the ability to do a careful, thorough review to I think a much greater degree than a conference can do.

What do you think should be the purpose of our conferences?

Conferences traditionally have been a way to get ideas out, get them out quickly, and be interactive. Workshops now seem to be taking on that role more than conferences. In part, I think that's because we have neglected journals as a discipline, and as a result, conferences became the primary way that we measure people for tenure and promotion in the field. So conferences have, therefore, necessarily become ever more strictly refereed, ever more competitive, and as a result, I think, the tone of conferences has changed a bit, and not necessarily for the better. We may want, at some point, to rethink our whole publication process, and recognize that some years from now, and probably not that many years from now, we're going to have all of our publications electronic: journal, conference proceedings. There will be nothing on paper, and that can change the way we do reviews, the way we evaluate publications, and the way we evaluate people for promotion. And then, I'm hoping that conferences can take on more of the flavor of a workshop: more interactivity, more half-baked ideas, half of which are no good, but the ones that

are good will then be much more valuable and much more stimulative than statements of things that are already done, packaged, finished, and presented.

"Being fully electronic is likely to change the nature of how we evaluate people". Why does being electronic change that?

The first thing it changes is how we publish. Right now, our publications are structured by the clock, because conferences occur at a certain time, and that drives that process. Journals come out in print in certain intervals, and have limits. There is a limit on how many pages we can put in a journal. We have that in TODS. Any journal I've been involved with has some kind of limit. In the electronic domain, there is no need for page limits, except as mandated by quality and readability. There is also no limit on the number of papers we can have, or when we can have them. There is no need to have, let's say, four publications of a journal per year, or one instance of a conference per year. These things can happen whenever something is good to put in there. And there can be some push notification for people who want to subscribe: "there's a new TODS paper, go look at it". Having that kind of dynamic mode of publication, we can now define the venues that we want to treat as the ones that count in terms of tenure and promotion. And that can be done independently of how we disseminate ideas quickly. So there are new possibilities there. I think it would be worthwhile for the community to think about what the end point ought to be, and then to back up from that, and to say how do we get from where we are to where we are going.

As a community, are our experimental results meaningful, believable, statistically valid, and repeatable?

[…] don't write a book, but I did. Try to stay focused on an area and establish yourself, but I didn't.

I am sure many of them are, but certainly, not all. But not always for, necessarily, bad reasons. If I look at some of the work that I've done lately, looking at architecture-aware algorithms for multicore, and cache awareness, when you move from one platform to another, the results are going to be remarkably different. If you simulate, rather than do it on a real machine, the results are different. Not necessarily wrong, but different. I think what SIGMOD is trying to do with repeatability is a good step in that direction. I would also like to see us focus more carefully on proper statistical methodology in our results. Sometimes it seems that we run an experiment, we like the results, and we write the paper. As opposed to stepping back and saying "Did we do a good experimental design?"; "Are there other experiments we can do?"; "Are our results truly statistically valid?". Having a spouse in the pharmaceutical business and seeing what they have to go through to convince the FDA to approve a drug, and the types of studies they have to do, we are not at their standards. Perhaps we don't have to be. But I would like to see a better delineation between careful statistical studies than some sense of definitive pieces of work, and perhaps, more cursory experimental work, that is there to stimulate new ideas. The way we have things structured now, where our papers seem to have an experimental section that has to be there, whether there really needs to be one or not, we're trying to be somewhere in between, and I think uncomfortably so.

You mentioned multicore architectures. Is that something that you've been working on?

I would like to see us focus more carefully on proper statistical methodology in our results. Sometimes it seems that we run an experiment, we like the results, and we write the paper.

Well, earlier, we were talking about being chair, and we went off on my discussion of teaching. One of the things that I've done less of in the past six and a half years as chair is research. I'd been very focused on that up to that point. I put a lot of effort into department building, and into the undergraduate curriculum. But having great undergraduates at Lehigh, I managed to do some work with an undergraduate on multicore architectures and their impact on database algorithms. That was a very nice experience. The student stayed on to do a Master's, and then went on to do PhD work at the University of Wisconsin, Madison. I've gotten really interested in how the changing computer architecture is going to have to change the way we think about database systems. We seem to be at a point where rethinking the engine could truly be the right thing to do. We talked about that in the Claremont Report, we're switching now towards at least thinking and maybe going to a column-oriented approach rather than a row-oriented approach. We have to think about parallelism… We have transactions, and if I'm going to be running thousands of transactions, I've got all the parallelism that I need. But to have sufficiently little conflict among transactions to get that parallelism, you then run into trouble with cache footprint. And so we clearly do have to start thinking about parallelizing the engine at various levels. I think that going forward, that is going to be a truly critical thing for us to address.

For our younger readers, can you explain about the cache footprint issue?

In the old days, all we cared about was disk and main memory. If I brought data from disk into main memory, I could do whatever I wanted with it, and it was effectively free, because disk was so slow. Main memory now looks like disk to a processor, or worse, because processors are so fast, and memory is relatively slow. And so, processors have multiple levels of higher speed storage called cache. The idea is that just as we brought data in from disk to memory for efficiency purposes, and tried to optimize that, we now need to think about how we optimize bringing data from memory into these caches. We have a similar issue, although, there are differences in it. While a database system has control over what it brings into memory, the hardware controls how cache is managed. So in some sense, the database system is pushing on a string, and doing cache management requires a different kind of cleverness. As a result, it requires a bit more heuristics, it is a bit less repeatable, but clearly, it is something we will have to think about. But it's not just cache. It's also the fact that we all have core duos, so we have a couple processors. We are going to be having hundreds of processors, and on the server's side, it is probably going to be many hundreds of processors. So we have this dua problem of keeping all

the many cores busy, but yet not requiring so much data that we wind up having cache misses, and now our cores are no longer busy, but simply sitting there waiting on cache misses.

Some people have argued that column stores are good for better L2 cache utilization, so would you agree with that?

I think that is a distinct possibility. I haven't seen the actual data yet to support that, but intuitively, what you want to be able to do is not have to bring in large rows. Most of which contain information that you either don't need at all, or don't need for this particular operation. Since caches are relatively small, you don't want to pollute them with stuff that you don't need, because it won't be there anymore by the time you may actually need it.

You commuted to Bell Labs from Pennsylvania for a long time. How long was your commute?

It was only 70 miles. I had mentioned earlier the two career issues, so my wife works west of Philadelphia, we live just north of Philadelphia, and when I went to Panasonic, that was about 45 miles. Bell Labs was longer, but I was able to telecommute once in a while. It was a wonderful commute: beautiful drive, a covered bridge, something like one-and-a-half-lane highways, deer, woods, streams. On the other hand, when you want to get back and forth, it does take a lot of time.

How long?

I did it once in an hour and twenty minutes. I don't think you would have wanted to have been a passenger on that ride. But I always treated it like a 2 hour commute from a scheduling point of view, so I wouldn't feel pressure to have to be there in a hurry and perhaps take unnecessary chances.

That meant your whole day was 3 hours shorter… Unless, did you find something useful you could do during that long drive?

Well, I'm supposed to say the nice intellectual things… ESPN radio and rock and roll. Sorry, I can't be inspiring there! What I did by trying to telecommute a couple of days a week was that I amortized my commute time over more days. For a lot of what I was doing, telecommuting was as effective as being there. While working with my database group, we sent email to the person in the next office! So, I could be home just as well. Also, the various business units of Lucent are all over New Jersey and elsewhere as well. We had interacted with people in the Chicago area, and some people in Britain for a while, and so we did a lot by teleconference, which I could do just fine from home. So I tried to schedule things strategically. Of course, when I moved to Lehigh, and then became department chair, I did need to be there every day.

How far away is that?

It is 35 miles. So I'd say I cut my commute in half, but it is not entirely fair because I have a little bit more traffic to deal with. I managed to make all that work. It is one of the compromises of the home and career balance, one that I think, overall, I'm very happy with.

You have two kids, or two teens. Are they following in your footsteps?

Thankfully, no. They have never had strong interest in computing. I'm not entirely sure why, although I love a story back from the preschool days, where the preschool teacher asked "what do your parents do?". "Mommy works with computers, and goes to meetings". Okay, well true, she does statistical analyses for a drug company. "Daddy goes to meetings". I knew something was wrong at that point. So, my daughter, while not being interested in computing, is very interested in mathematics. I was an undergrad math major at Williams, so you could say that's in my footsteps, but I've not played up by being a math major. Whether she will go into mathematics for a career or not remains to be seen, but she is strongly interested in math, and the sciences. So, in some sense, yes, but differently[2].

Do you have any words of advice for fledgling or midcareer database researchers?

I mentioned earlier, don't write a book, but I did. Try to stay focused on an area and establish yourself, but I didn't. I worked in transactions, I worked in relational database theory, I worked in object-oriented databases. So a lot of the traditional advice, I've not followed. And so I think, giving advice that I can be more credible giving, would be, take advantage of the opportunities that are there. They are going to be different for each individual. Opportunities don't necessarily come back, and they don't necessarily show up at the optimal time or when you want them. And so, take advantage of it, if it hurts you in some ways, it's going to help you in others. And enjoy what you're doing. If you enjoy what you are doing, you are going to be much more productive, and much more successful.

Among all your past research, do you have a favorite piece of work?

Oh, that's a tough one, because I have worked in so many different areas! I guess if I have to pick one thing, I gotta pick my one JACM paper, even though that goes way back to a 1983 issue of JACM.

What was that paper about?

That paper was on my work on transaction processing where I took the type of multi-granularity locking used in System R, and extended it to have an arbitrary set of fundamental operations, instead of just read and write. Something that I think is still relevant in many ways, and may actually become more relevant if things like transactional memory start to come into being in a wide-spread way, where you can truly have multiple levels of atomicity, and then looking down at the lower levels, those are like harder operations.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

Go back to playing ultimate Frisbee with the students again, like I did at the University of Texas.

[2] She's graduating in May 2013 from Harvey Mudd with a degree in engineering.

It's not too late, now that you're not department head, you can go out with those undergrads.

I don't have the knees for it anymore. My kids outrun me. My son outruns me by about a factor of two. So, I think I've reached a certain age.

If you could change one thing about yourself as a computer science researcher, what would it be?

I think I would find a way to give myself more time to write code, and build real systems. I started off on the theoretical side, working under Jeff Ullman. Jim Gray taught me about database systems as systems. He is certainly one of my greatest mentors there. I have always been somewhere between theory and systems, bouncing back and forth between them. I really wish I could have had more time to get involved in system building, and actually be deeply involved in the actual construction of a major system.

I remember Rafael Alonso talking about having the web in the palm of your hand. But why would anybody want the web in the palm of your hand? Well, it is now a joke...

Do you mean sort of on a Dewitt model?

Well, that's a bit of an extreme, perhaps. But that is what I am referring to, yes.

In one of these interviews, he said that his students won't let him touch the code[3]*! So you think these people are out there coding up all this stuff, but maybe they actually aren't.*

Maybe not, but certainly I think I have been less hands on personally than I would enjoy. The reason I went into computer science originally was that I enjoyed programming. Then I found out that it looks more like mathematics. Well, that's okay, I was a math major, I can go with that too.

Kind of funny that way, isn't it?

I have been looking for the balance ever since.

Well, thank you very much for talking with me today.

Thank you, it was a pleasure talking to you.

[3] David DeWitt speaks out: on rethinking the CS curriculum, why the database community should be proud, why query optimization doesn't work, how supercomputing funding is sometimes very poorly spent, how he's not a good coder and isn't smart enough to do DB theory, and more. SIGMOD RECORD 31(2): 50-62, 2002.