Jennifer Widom Speaks Out

on Luck, What Constitutes Success, When to Get Out of an Area, the Importance of Choosing the Right Husband, Outlandish Vacations, How Hard It Is to Be an Assistant Professor, and More

by Marianne Winslett



Jennifer Widom
http://infolab.stanford.edu/~widom/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the Department of Computer Science at the University of Illinois at Urbana-Champaign. I have here with me Jennifer Widom, who is a professor of Computer Science at Stanford University. Jennifer's research currently focuses on data provenance, management of uncertainty, queries over web services, and data streams. Jennifer is a member of the National Academy of Engineering, is an ACM Fellow, and is a former Guggenheim Fellow. Before joining Stanford, Jennifer worked at IBM Almaden Research Center. Jennifer's PhD is from Cornell. So, Jennifer, welcome!

Thank you!

Jennifer, you changed the name of your group from "The Database Group" to "The InfoLab". Is that because database research is dead?

No, it's not, of course. I think of myself, actually, as doing core database research. Most of my students do core database research, and will for a long time to come. But we do have students in our group who work outside the database area. The name change was primarily so students didn't get pigeonholed inappropriately. For example, we had a student who did a thesis in photo browsing and labeling. We didn't want that student to be labeled as a database student. We have students who do information retrieval; again, we don't want them to be pigeonholed as database researchers. Database research is a subset of the work going on in the InfoLab.

You seem to have a knack for picking up-and-coming research areas and being among those who lead the charge. What is your process for choosing problems to work on?

More or less random! When we started working on the LORE project (a database management system for semi-structured data), it was a small offshoot of a data integration project that was using a semi-structured data model. I said, why don't we build a system to store and query that data? And we did, and it grew into a big project, eventually becoming a database system for XML.

I also had a project on data streams. I had wanted to work on data streams for a long time; I thought it was an interesting new model, but I couldn't convince any students to work on it. Finally I got a couple of students interested and we launched a project. Since then, the data stream area has become very popular.

As for my most recent project, I had been thinking it was time to have a new research direction, but I really didn't have any ideas on which way to go. I was doing my morning jog one day when I started to think about uncertainty and data lineage, or provenance, and how they seem to work together in a lot of applications. I decided to build a system to handle these two aspects of data, together with the data itself.

So I would say all of these research directions are chosen fairly randomly, almost as knee-jerk decisions, without a major thought process, or a huge vision. I don't think of myself in fact as a visionary whatsoever.

How do you decide when it is time to leave your current area of research and move into a new one?

I probably do leave areas fairly early, in a sense. I definitely leave and move on to something new when there is still lots of work to do in the old area. There are a couple of driving factors in play here. The first is graduate student interest. Suppose that I have been working on a project for several years and I have a fairly mature prototype. Then even if I have a list of five obvious thesis topics, if I get a first year PhD student, they won't want to work on one of those topics. The area is now four or five years old, and the students don't want to work on a well-defined topic in an "old area". So the students themselves drive approximately a five year cycle in my projects.

The second factor is that if an area gets to have lots of people working in it, I prefer to move on to a newer area with fewer people working in it. I like to do things early and then move out. Sometimes I think of an analogy of surfing: you're riding a wave and then at some point you just cut out and let the wave continue. I like to do that.

Do you have any tips for us on starting a research project in a new area?

To start a project in a new area, I recommend that you spend a year or so on foundations and figuring out how the new area fits in the scheme of things. If you are working in a new area, you can spend a long time working out foundations. I think foundations are important, but sometimes you just have to say that it is time to push on and start building a system, looking at more practical issues. Get some applications, some data, and work with those.

What do you view as your most successful research project to date?

I think that a project of mine is successful if people in industry get interested and start building similar things. Right now I feel that the data streams project has been the most successful. Industry has gotten very interested in the area. People are interested in what we did in the project, and in the query language we developed.

The other project I feel most strongly about is the LORE project, where we built a system that was eventually used for XML. That project got very well known because of XML, and I think it was lucky timing. We had a database system we were building for a semi-structured data model, and along came XML, which was very close to our model. We switched our model to XML, which didn't take much effort at all, and suddenly we had a system for XML very early. So I think the LORE project was also quite successful, and some lucky timing was involved in its success. Lucky timing is not such a factor in the success of the streams project.

So are you saying the industry is taking its cue from what's happening in academia?

No, I don't sense that's happening. I think industry finds business needs, and then they look for the technology that meets those business needs. So that's what I am finding in streams: people from different areas are saying that they need technology that looks like data streams, looks like continuous queries. They discover the technology after they identify what kind of technology they need.

What about what David DeWitt says, that we don't need specialized databases for stream data management?

That may be true, and I'm not going to argue that we absolutely have to have a special purpose data streams system. Our project produced a data stream management system, built from scratch, and that was a lot of fun, but is that going to be the way data streams come to the mainstream, so to speak? Not necessarily.

There are companies building systems that are basically what ours was, a native data stream system. There are companies whose product is based on data streams and continuous queries hidden inside the software. And I have very little doubt that the major DBMS vendors are going to add stream technology to their systems. Other technologies in databases have gone this way, too. At first, little startups pop up, building native systems. XML databases went this way. Object-oriented databases went this way. I am sure there are many other examples. But then the big vendors said, "Hey, we can add that to our systems. It's not that hard, it's seamless with what we already have." Then the little companies get snuffed out. I wouldn't be surprised to see that happen with streams technology. I think the verdict is still out, so I'm not going to argue with the claim that we don't need a special purpose data stream management system.

A different argument might be that we don't need data stream technology at all. I disagree with that. I think that in the last year or two, industry has shown that you do need a different way of looking at streaming data, and you need continuous queries.

So what's the killer application that has emerged from industry?

Financial monitoring is one of the major applications right now. Another is web click data streams, which tell where people are going on the web, and doing things with those streams in real time. There is an application called business activity monitoring, where people want to record everything that is going on in their business in a streaming fashion, and have dashboards where they can view it and allow their high-level executives to make decisions. Those are three real applications that have a real need for this kind of technology.

Where do you think the field is going now?

I can toot my own horn and tell you about my next project, Trio. I'm interested in managing uncertain data, and I do think people are going to want more and more support for uncertainty pushed into the DBMS.

People have worked on uncertain data for a long time, but it hasn't been a primary focus. When I look around now, people are ready to put information about the certainty of their data into a database and start querying it. That really hasn't happened yet, but when you talk to people, they often tell you that they have data that isn't black and white. So that's what I am interested in right now.

This topic seems to connect well with the area of lineage, because if your data isn't certain, you might want to know where it came from and how it was derived, to try to figure out the quality of the data. Quality is very important.

Now I'm done with tooting my own horn. In terms of other areas, data integration will continue forever---people will always work on some problem in data integration. So the database field is always going to have a big component of that. What exactly people will try to do, I'm not sure. I have one student looking at web services and trying to integrate queries across web services.

Data sets are going to get bigger and bigger, messier and messier. Data cleaning is important.

Jennifer, you have spent your whole career in the heart of Silicon Valley, and you've never done a startup. Why is that?

I'm not a startup kind of person. Let me elaborate on that a little. We did come close to doing a startup with the LORE project. But I found that going to meetings with funding people and potential CEOs was not a fun thing for me. It was fun at first, when it was novel, but I lost interest at some point. I also like to be in control of my time and be my own boss. Even if you are the boss of a company, there is always somebody else that is effectively your boss: your funders, or your customers. I wasn't too excited about that. I also like to have predictability, and startups are very unpredictable. For some people, the unpredictability is the joy of doing a startup. For me, that's not a joy. So I've been very content with consulting and sitting on advisory boards. For me, that's the perfect interaction in Silicon Valley. Doing some consulting and talking to people makes me feel that the research I do is grounded, without my having to go for the whole startup package and all that that entails.

You're one of the few people I've interviewed with kids in grade school. How can you be so productive and raise two kids?

I think the most critical factor by far in managing children and having a career is to have the right husband. My husband is also a professor, but we are very, very symmetric about everything, so I have not taken on more burden than him in child raising. So, for those of you watching or reading this, think about that before it is too late!

Having the right husband is the primary factor, but the next most important factor is being efficient, knowing what's important and what's not important, and not being shy about ignoring those things that are not important. Or perhaps it is not *ignoring* so much as *focusing* one's time and energy on what really matters at work, so that you free up time for your family.

How many hours of sleep do you get a night?

I have been known not to get quite enough sleep sometimes. In busy, busy periods I'm a 5-6 hour sleeper. It is definitely important for me to have those extra couple hours during each day for managing the family and my work.

How do you manage to keep your desk so neat?

Those people who have been to my office know that my desk is very neat, and I really love throwing things away. I am a "thrower-awayer". A lot of people are pack-rats, and I am the opposite. I have a philosophy that if you throw everything, or nearly everything, away, the amount of time that you are likely to spend reacquiring something that you threw away and then found that you needed is much, much lower than the amount of time you might spend doing things with those things that you didn't throw away. To put it another way, the time invested in dealing with the case where you accidentally threw something important away is quite low. The time invested in dealing with all the stuff you thought might have been important is quite high. I throw away things in the office, most everything, and even on the computer I throw things away. So my computer desktop is fairly neat.

Last year at SIGMOD, we had a keynote talk called "MyLifeBits, A Transaction Processing Database for Everything Personal," by Gordon Bell from Microsoft (http://research.microsoft.com/barc/mediapresence/MyLifeBits.aspx) . That talk was about the trend to record absolutely everything about one's life, to keep it and have it available. So, for example, let's keep a record of yesterday's phone conversation, just in case we need to look it up someday. I do not want that at all. It's completely against my philosophy to have all that stuff, even if it is supposedly easy to access, because then you spend time trying to access it, and I think that is not a good use of time. I put that all into the efficiency category that I was talking about earlier: trying to live an efficient life so that one has time to work and be with one's family.

Your undergraduate degree is in music. How does a music major end up doing database research?

I was a trumpet performance major at the Indiana University School of Music, where we had the whopping requirement of taking *three* classes that were not music theory or music performance. One of those classes I took was even in the music school, and was called "Computer Applications in Music Research." I chose the course completely randomly as one of my electives. We wrote SNOBOL programs to analyze streams representing music, and I got hooked. It was my junior year in college, and I started taking some computer science classes at Indiana. I finished my performance degree in trumpet, and then I stayed at Indiana. I continued actively in music there, but I switched to a master's program in computer science. That is really how I got my undergraduate education in computer science---I like to think of it that way. They admitted me to computer science at Indiana on the basis of a few classes, and then I really got some breadth, and also got into some research to some extent. After my sort-of-bachelor's degree, which is actually my master's, I decided to get a PhD and moved to Cornell. So that's how I moved from music to computer science. I continued playing my trumpet until 1992, which was quite a while after I finished my PhD.

What made you give up the trumpet?

I was tired of practicing. It's a little bit like a sport, though some people don't realize that. I was practicing my trumpet an hour and a half a day during the time I was working at IBM Almaden, and playing actively in musical groups around San Jose. One day I realized that I didn't really want to do all this practicing any more. I didn't want to just rachet back, because playing the trumpet is such a physical activity. So I just decided to stop. However, I am thinking about taking it up again because my son has just taken up trumpet and he needs someone to play duets with.

You have been both at an industrial lab and at a university. How do you look back on your days at IBM?

The days at IBM were great. They were very easy days. At that time at IBM, we were really just chartered to do research. There weren't a lot of administrative duties; you didn't have to get grants, like a faculty member would have to. We really had a lot of license. I was in a group working on the Starburst project, building a big prototype that was excellent infrastructure for trying out research ideas. I spent most of my time doing research; I believe I really established my research track record during that time at IBM. So my days at IBM were very idyllic.

I also learned about databases mostly at IBM. My PhD is in programming languages, so it was a time to learn about databases, have a lot of freedom, in what was at that time really one of the greatest groups in the world. So it was a great five years I spent there.

What led you to make the switch from industrial research labs to academia?

I am a child of a professor; I always thought being a professor would be a great thing to do. With academia in my genes, the opportunity to have a faculty job at Stanford was something I couldn't turn down.

Would you ever consider moving back to a research lab?

Nope. I love being a professor. It is the greatest job on earth. I believe that to be true.

How does the programming languages community differ from ours?

It's very different. I got my PhD in programming languages and went to some of their conferences for a few years. I find the database community to be friendlier, more social, less self-conscious or posturing. I don't want to put down the programming languages community, but I really find the database community relaxed. I think it may have to do with the funding situation. Now funding is hard to find no matter what community you are in, but there was a period when database people had it quite a bit easier than those in other fields. Also, the database community has a stronger connection to industry, a lot of self-confidence that what they are doing matters to people, and that may be less true in programming languages. There are a lot more party animals in databases, too!

You like to take exotic vacations with your family. (When I was preparing for this interview, some of my informants used words like "outlandish" and "dangerous" to describe your vacations.) How did the vacation turn out where you were going sailing in Thailand with your husband and kids when the tsunami was on the way?

There are a few misconceptions in your question. I'm not going to argue necessarily with "dangerous" and "outlandish"---well, we would not put our family in danger, but perhaps some people think our trips are outlandish. We do take exotic adventure vacations. Another misconception is that---I don't think anyone knew the tsunami was coming, did they? I think it just came. The third misconception is that actually we weren't in Thailand for the tsunami. We were trying to make a reservation to charter a sailboat to sail around some islands in Thailand for that particular time. The sailboat we wanted wasn't available, so we went to New Zealand instead. In fact, we were white water rafting in New Zealand when our guide mentioned to us that a big tsunami had hit in Thailand. We thought, wow, we could have been sailing there! But we weren't, we were just fine. We went back to Thailand the following year, chartered a sailboat and checked out where the tsunami had hit.

Jeff Ullman has a company that makes a system that automates the generation of homework. Do you like using his Gradiance software?

I am a huge fan of Gradiance. (Actually, I argue with Jeff a lot about some aspects of the system, but in reality I love it, so put that in print.) Gradiance generates homework problems from a questions bank, and then it corrects them and gives feedback, all automatically. Students can do their exercises over and over, getting different instances of questions each time, and get it graded immediately. The students love that. Gradiance also has a SQL engine, for introductory database students. You can give Gradiance a schema and data, and assign queries in English. The students write the queries in SQL, and the queries are run against the database, providing immediate feedback. Gradiance will tell the student that they got the wrong answer, and show them the data and the correct query answer. Then the student can try to write the query again. Gradiance has some very clever techniques, like having a second hidden database so students can't fool the system once they see what the query answer is. Students really react positively to the SQL engine

and what it offers. So I have found Gradiance to be an excellent teaching tool, one that I really enjoy using.

I have heard that you think presentation skills are extremely important. Can you expand on this point?

Regarding oral presentation skills, in our group we do have a reputation of having our students give practice talks and then ripping them apart. I think we have very high standards in our group for what it takes to give a good conference talk, and how important it is to give a good talk. So our students do give talks over and over until they get them right. And we do a lot of coaching.

I feel the same way regarding writing skills. I think that writing a clear paper is extremely important and very difficult. We spend quite a bit of time talking about what constitutes a good paper and what doesn't. I am very fussy with my students' drafts. I always warn my younger students that the first time they give me a paper and I mark it up, the student will barely be able see the black ink underneath the blue ink from my pen.

One of my senior students has decided that his papers are *too* well written. He thinks they are so well written that referees have a much easier time finding something to find fault with, because they actually understand the paper. So he is thinking about conducting some experiments where he takes a well written paper, makes it less well written, and submits it to conferences to see if he gets fewer complaints. He really believes this to be the case, and I think there could be some truth to that, unfortunately.

You can get the benefit of the doubt sometimes if the reviewer is not sure what you're talking about.

Exactly. I think that happens quite often. The faster someone reviews a paper, the more it works to your benefit to have not done a good job with your paper.

What's next in your career path? Do you envision yourself as a dean?

At some point way back, I thought it might be fun to be dean. Now I don't think it would be fun, for the same reasons as for a startup: when you are a dean, you start to lose control over things that are important. You lose control over your schedule, you have to dress up all the time (I am not big on dressing up), and you are interacting with people to try to get big donations and such things, which I don't think is my cup of tea. As I said earlier, I am not a visionary, and I think deans ought to be visionaries. So I don't see a deanship as something for me.

Actually, next in my career path, my family is going to take a 14 month trip around the world and not work at all. When I come back from that, I guess I will see how I feel about things.

Do you have any words of advice for fledgling or mid-career database researchers or practitioners?

I can comment on researchers primarily. It is no secret that it is very difficult right now to be a young researcher. I don't think people should pretend otherwise. It is harder now than it used to

be. I think you really have to want it with a passion if you want to be an assistant professor. You have to get grants, which is harder than it used to be. You have to make sure you get major publications, which is harder and more random than it used to be. So it's not easy, and you really do need to want it. My advice is to get that fire going, hit the job with a passion, and don't get discouraged.

I guess "don't get discouraged" is the best advice. You have to look at the bigger picture. If your papers don't get into one conference, it's probably not because you are a terrible researcher. Just wait for the next conference and try to look at the long term, try to make your work have overall impact rather than worrying about each specific instance.

It sounds like you think the acceptance rates at major conferences should be higher.

It's possible. We could go into a whole discussion about conferences and publishing and what's wrong. I think there is a problem, and there has been a lot of discussion over the last couple of years. I'm not sure what the best solution is. Some people have talked about online journals with no acceptance rate, just a pure quality threshold, which I think is an interesting idea. It is a very complicated issue, but I do think that right now in our very selective conferences, many valuable papers aren't being accepted. I do worry about young people's careers because of that. Students also have high pressure now to get papers published, if they are going to look for academic jobs. So it's tough right now.

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

I don't have enough time to learn about other areas of computer science, or areas outside computer science. (I suspect that this is the most common answer to this question.) I would love to know much more about all kinds of things. Even closely related things, like information retrieval, data mining---I mean, these are practically in my field and I don't know enough about them. AI, natural language understanding---these are all things I ***should*** know more about. And then there are things I just ***want*** to know about. I would like to know more about biology because it's so popular and interesting, graphics, just all kinds of areas.

Having time to learn about those things would be great. I don't foresee it happening. Maybe when my kids go to college.

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

I would probably like to do more coding. I think of myself as a systems person, primarily, and my students all build systems. I would like to move down more in my level of interaction with those systems, because I am really at a high level. I would rather know more exactly what is going on inside the system, and even participate myself in building it. That would be great, but I don't have time for it now.

Thank you, Jennifer, for talking with me today.