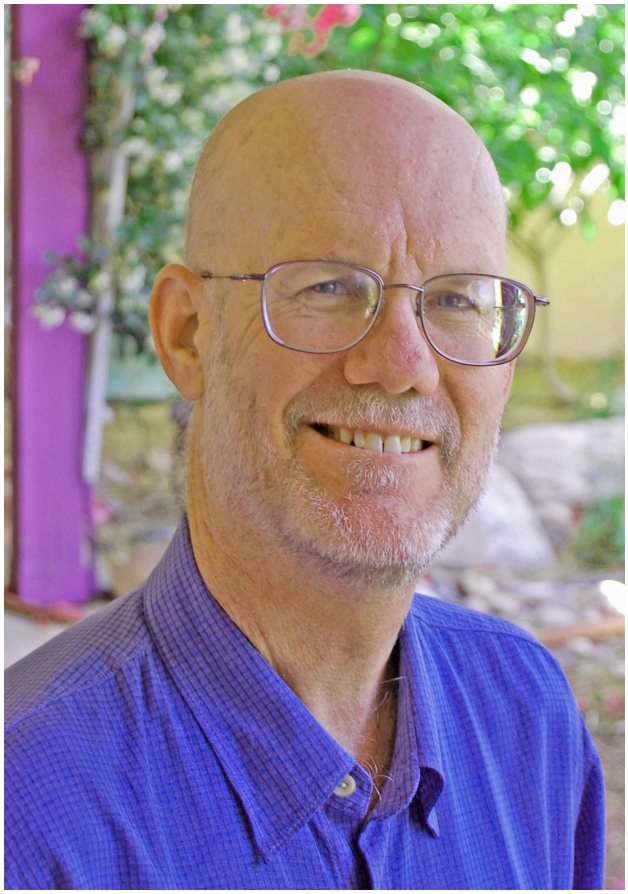
Rick Snodgrass Speaks Out on Standards, Personal Brands and Science

Marianne Winslett and Vanessa Braganholo



Rick Snodgrass
http://www.cs.arizona.edu/people/rts/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we're in Phoenix, site of the 2012 SIGMOD and PODS conferences. I have here with me Rick Snodgrass, who is a professor of computer science at the University of Arizona. Rick has served as the Editor-in-Chief of ACM Transactions on Database Systems, the chair of ACM SIGMOD, the ACM Pubs Board and the ACM History Committee. He has received the SIGMOD Outstanding Contributions Award and ACM Outstanding Contribution Award and he's an ACM Fellow. Rick's PhD is from Carnegie-Mellon University.

So, Rick, welcome!

Thank you!

Rick, you are best known for your work on temporal databases, and you even worked hard to get temporal constructs into the SQL standard. What role do you think database researchers should play in the standards community?

I think it's very important for the database researchers to have a role. Unfortunately, by the way that the Standards Committee is set up (it's run and funded by vendors), it's very hard to spend time with them. They're open to having people come in, but it has to be funded by the researchers. So I would actually like to have the committee be more accepting of people from our research community and actually pay for them to come in and invite them in. For example, the ANSI standards are closed. They don't even release them to the community for comment until they have been finalized. I think there should be much more of a dialog with the research community.

There is really no excuse for not having an accurate and complete provenance on the ideas that you're working on.

Are there "mistakes" that we could have prevented if we would have been more involved with that?

The standard is a very big standard. It is thousands of pages long. I think perhaps had we been involved, we could have found more foundational aspects to reduce the redundancy in the standards. So that would have been one place. I don't know if that is a mistake, but certainly the more input is better, I think.

A standard that is thousands of pages long? It sounds like almost contradicting terms to me. How can anyone understand?

I don't think anyone totally understands. I think it's about four thousand pages. I think the standards body (the people that did it), who have been working on it for decades understand it very well. Us normal mortals, I don't think so.

Isn't there a gap between what the standard says and what people actually implemented?

Yes, so it's much bigger than what most DBMSs implement. Although each DBMS implements a portion of it, of course there are a lot of inconsistencies between the standard and the implementations.

So in what sense is it a standard if people don't implement all of it and then the parts they do implement are inconsistent?

Well it's better than having a free-for-all where they are all different. So there is some compatibility. So we should be happy for that.

So I guess in industry there are incentives for diversity. That people can't switch between products so easily. What are the incentives for standardization?

The standards process is very interesting because you have all these vendors -- who are competing with each other -- getting in the same room and trying to come up with something that is a standard. That really shouldn't work. So it's amazing how well it does work, given that situation. They are all very much competitors and they each want to win. So the fact that we get anything at all, I think we should celebrate.

People are saying that the relational temporal database constructs don't apply to graph databases (for example, in biology). Do you agree with that?

No, I don't. Actually I've done some work with temporal XML graph data. I think the underlying concepts, like sequenced, apply directly. Well, databases have foreign keys and that's a lot like a tree structure also, or graph structure. So I think that if we get to the foundational concepts, hopefully it should apply, with modification, anywhere.

What parts need to be changed to move to the graph databases?

I think the ideas need to be applied to that specific place. So when we did our work with temporal XML, we had to figure out what were the unique aspects of XML, but there were actually very few. If you have an XQuery query on XML, you can say "I want to evaluate that on a temporal document at each point in time". That is what *sequenced* does. So that applies directly. Now, how you actually implement that efficiently is a complex problem, but at least you know what the semantics should be.

You have a project right now on ergalics. What is that?

Ergalics is from the Greek word "erg" which means work or tool. And so it's taking computational tools like DBMSs or compilers, or other kinds of tools. It's a science of computational tools. The reason we needed this word is because computer science has the word "science". So the science of computer science is a very awkward formulation. So I came up with this word, "ergalics". I did a Google search and no one had used the word, so it's a new word.

Computer science really has three different perspectives in it. One is mathematics… so dependency theory, asymptotic complexity, etc. Another one is engineering. Most of what we do is engineering. Engineering is doing something better, faster, cheaper. So you're always trying to do a better job. Most of PODS papers, for instance, are in the mathematical perspective; most of SIGMOD is in the engineering. We have actually very little science methodology or predictive theories. That's where I think we need to also go. That does not mean that the other two perspectives aren't just as valid. Mathematics can oftentimes inform science and science can inform engineering. Right now that centerpiece is not there. So we jump from mathematics to engineering. I think that really good engineers have intuitive understanding of the predictive rules by which these computational tools work, but we haven't yet articulated those.

So what would they look like in the case of databases?

So I've been working on that for quite a while. It's very difficult research in that whenever I find something I want see "why is that?" and then I look in the code. The way I like to think about this is to treat a DBMS like a biologist would treat a rabbit. So a biologist looks at a rabbit and says, "this rabbit has big ears". Why might that be? Well maybe it is because they need to hear better. Well, but there are other animals that need to hear better that don't have big ears. So why do rabbits? Especially in the desert? Desert rabbits have the biggest ears. So maybe it's because they need to release heat. Well, then you can do studies and experiments, in that case, blood flow in the ears could make a difference. A biologist does not say, "How do I make a rabbit run faster? Maybe if I make the legs longer…". That's not a question you ask. You ask "why". Why is the rabbit the way it is? So I'm looking for predicative theories about DBMSs that would apply across DBMSs. So, SQLServer, DB2, Oracle, Teradata…these are implemented by different people. They are totally different products. What are things we can say about them in general?

Okay, but rabbits are evolving, but slowly. These DBMSs are evolving super fast; at least we hope they're evolving fast…

Actually most of their foundational parts have been around for close to 30 years. I mean, cost-based query optimization, locking, concurrency control… So actually, there is a lot that has remained pretty steady. And so we can ask questions about them. For instance, if you add a query operator, a relational operator, what will happen to the number of optimal plans?

You mean a new kind of operator? That's like adding a fifth leg to a rabbit.

This is informing the engineering. So right now we would say, "We want to make some queries run faster". So I'll add a certain kind of index or I'll add another kind of relational operator. That has the benefit of making some queries run faster. So that's good. It also makes the optimizer more complex. So we're going to have more mistakes: sub-optimal plans. So do you get more advantage or disadvantage? Is there a point in which when you are adding an operator, on average you actually slow down the DBMS? That's a question that you really can't ask about a single DBMS, you have to ask it across the class.

And what's the answer to that one?

I'm still working on that one. So we don't know that. That's a very interesting question for engineering. That's just one of many.

Well at four thousand pages, maybe we're approaching the point where the next thousand pages would just bring it downhill.

Yes, as a matter of fact one of the complaints against *ergalics* is that well, these are programs. We understand programs. I mean we can look and see exactly how they work. Well a DBMS is so complex, no one understands it. Science deals with the universe, which is also very complex. They have a whole bunch of methodologies that we can use to understand these very complicated things.

So can you give me an example of an insight that you've gotten to date, using this approach?

It looks like as you add operators, you do have an increase in sub-optimality. So it looks like there is a limit at which point, we get diminishing returns and it actually goes the other way.

Do you know yet whether we've already reached the limit?

I haven't gotten the theory to the point where it's that specific yet.

In the database research community we don't often think about branding, but I know that you do. So I guess I'll start with a little story. Jiawei Han told me that we needed to change the same of our research group at Illinois and we did. We used to be called the Database and Information Systems Group, and now we're the Data and Information Systems Group. Jiawei said that if we called ourselves database group, everyone who is not in this area thinks that databases are a solved problem and therefore we're not really doing anything interesting. So this may be an example of a branding issue. So what do you think we need to do in the database community in terms of our brand?

So I think brands are very important. Since we're in the southwest we first need to define what branding is. It's not what you do to cattle to identify them. We're not talking about burning signs on their hides. We're talking about identifying in the minds of people what does this discipline or what does this person do. My wife is a marketing professor so that's how I know about branding. Disciplines have brands, departments have brands, universities have brands, and people have brands. And I think it is very important for people to think about what their brand is. As far as a discipline, I totally agree with Jiawei that DBMSs are viewed as a solved problem. We're still trying to do better, we're still trying to do more engineering, but DBMSs are wonderfully efficient and powerful tools right now. So going from DBMSs or databases to data I think is exactly the right place where our discipline needs to go. We really understand data very well and I think that that is a skill that the world needs, scientists need for their data, engineers need it for their data, and even experimental mathematicians need it for their data. We have a lot to give the whole world.

So we'd be the Data Management Research or Information Management Research?

So I see information as being data with insight. You're adding insight. There is a hierarchy from data to information to knowledge to wisdom. I think we are experts at the first couple of levels. You get in the philosophy when you get up to the wisdom part or morality or whatever, but certainly those first few steps are very important. I think we have a lot to add.

So do we need to change our current brand or is it already in the right place?

Well this is the Special Interest Group on Management of Data. It's not Management of Databases. So yes, SIGMOD I think has the right name. As opposed to for instance IEEE Data Engineering. Well, that's kind of restrictive. They're only looking at a third of the thing, whereas we can look at the mathematics of data, the science of data and the engineering of data.

Well all those people out there who would say, "I'm in the database group". What should they be saying instead?

I mean if they're interested in databases and doing that, that's fine! I don't think that us as a society, but also as a general discipline should necessarily limit ourselves that way.

So a new name is called for?

Or just emphasizing MOD "Management of Data" and going back to that.

> ***A biologist does not say, "How do I make a rabbit run faster? [...]". That's not a question you ask. You ask "why". Why is the rabbit the way it is?***

What about personal branding as researchers? How should we handle that?

So I wrote a paper[1] with my wife, Merrie Brucks, on this, with a whole bunch of different approaches for personal branding. The goal here is to come up with a single phrase that when people hear that phrase, they think of you. And when they think of you, they think of that phrase. So, in my career, I've branded myself as "temporal databases". So when people need a temporal database guy on their program committee, they think of me and they think of some other people. And when they think of me, they think of temporal databases. How have I helped that? I've written glossaries, I've written surveys, I've written papers, most of which have the words "temporal database" if not in the title, then in the abstract. I'm now going towards "science of computer science". That's what I've been working on the last few years. That's an awkward phrase, so I

[1] Richard T. Snodgrass and Merrie L. Brucks, "Branding Yourself," *ACM SIGMOD Record* 33(2):117–125, June 2004.

came up with a new word. If you Google "ergalics" you'll see my name. And eventually when you think of Rick Snodgrass, maybe you'll think of ergalics. A lot of people have a problem with this because they see it as limiting, but you get to pick what you want to be associated with you.

Well then let's pick on Jim Gray because he is not here. So what is Jim Gray's brand? He's arguably our most successful researcher.

I would argue his brand is "integration of research approaches". He talked to everyone. He knew pretty much what everyone was doing and he could help them figure out how they fit into the big picture and he could articulate that big picture. So that's what I see his brand is. So a brand can be a topic, it can be an approach, it can be a methodology, and it can be a special ability like Jim Gray's…

Okay. Can we pick on some other people who are maybe a little more here? So what's David Dewitt's brand?

David Dewitt is very articulate and controversial. When he says something, people want to hear what he says. And he's brilliant so he's really good at bringing problems to the community and bringing solutions, and bringing places where we are not doing very well, which is very helpful for us. He's a fantastic person to be on a panel for this reason.

I think one can go too far and just spending all of one's time reading and one can go too far and not read anything. It requires some real skill to figure out where that middle ground is.

Oh yeah, he's exciting to listen to.

Because he can identify these issues. He did a talk on "Database Systems: Road Kill on the Information Superhighway?[2]". Perfect for telling us how we missed the boat in terms of the web. The web is a big database, but that's not how it's viewed.

That's how we view it, but that's not how the world views it.

That's right. We are road kill on this.

Okay, reminds me of the fact that you're in the ACM History Committee.

I'm no longer; I went off about a year ago.

Okay you were on the ACM History Committee. It's now history that you were on the ACM History Committee and computer science is all about, in my mind, inventing the future. So what role does history play in a community that is all about creating a new version of history, so to speak?

I have a couple of different responses. One is that, as I said, databases is fifty years old. That means that the people that started it are getting very old. So we're about to lose a lot of our history. The history committee has commissioned a lot of interviews with these early pioneers. For instance, Charlie Bachman was interviewed by SIGMOD[3]. SIGMOD paid for that interview by a professional historian. Everyone should read his interview on the ACM Digital Library. It talks about IDS (his original system), which was an amazing system that has a lot of similarities with the most recent systems. It was a main memory database system that used virtual memory, for instance. Pretty amazing. We wouldn't have that without a history committee in ACM. We need to grab our history before it goes away. We're always looking into the next five years. So we're going to be losing this very vibrant history that we have. Now we're not like physics and mathematics which has hundreds or thousands of years, but that's good, we can actually capture that. And we can put it in the digital form, using the technology we've invented.

I think we need to re-print Charlie Bachman's interview in the SIGMOD Record.

Well, it's 162 pages long…

162 pages?!

Yes, it was a two-day interview. I have to tell a quick story. I got a call a couple years ago from Charlie Bachman saying "Why don't you come over to my house? I just finished the interview". So I said, "Well, where do you live?" It turns out he lived a mile and a

[2] Keynote on VLDB 1995

[3] Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona. In: Proceedings of the ACM Oral History Interviews, 2006. DOI: 10.1145/1141880.1141882.

half from my house in Tucson, at that time. So I went over there and Tom Haigh was there. He had just finished the second day of the interviews. Charlie walked up and said, "Would you like some nuts?", and held out a little tray. I realized that that was the Turing Award Bowl...

Whoa!

So I got to get a nut from the Turing Award Bowl from Charlie Bachman. But you mentioned Jim Gray. Jim Gray is no longer with us. We didn't ever interview him. So that's a loss. And Ted Codd is no longer with us. So what you were doing through these interviews, with the other people you were interviewing, is going to be very valuable 20 or 30 years in the future, as well as now.

I decided to go back to the first ones and see what people were predicting and how much of it has come true, so we can compare our historical predictions against reality…

I bet you we are pretty bad at that.

I don't know. I don't know yet, I'll check it out.

So speaking of changing overtime, how has ACM's publications changed over time?

Oh excellent question. If you go back, say, 14 years… at this conference in 1998[4], I think that was in Philadelphia[5]. So if you went to that conference, you got a bound printed version of the proceedings. You took that and the other ones you went to at that time, and you put them on your shelf. I'm sure you had a shelf full of proceedings because that was how you got papers. If you didn't have it in your office you had to go down to the library to get it -- fourteen years ago. So SIGMOD, our community, decided to scan all of those proceedings and put them in digital PDFs. Not only that, they talked to all the other database societies and helped pay for, but also encouraged them to scan theirs. Five years later, SIGMOD gave to all of its members the ACM SIGMOD Anthology[6], two DVDs with 150,000 pages of database papers. Then SIGMOD went to ACM and said, "Other SIGs, you should do the same thing". So the SIGs paid for digitizing the entire past history of all of their conferences and journals. That formed the ACM digital library. Then IEEE (we had already done Data Engineering because SIGMOD worked with them) decided to digitize all of their past. So because of SIGMOD and the SIGMOD dues, which helped pay for this, all of the computer science papers are now digitized.

Well you would like that if for no other reason than its history, but if you look at the accesses to the library, which I've never done, how much do people look at that older stuff?

I'm not sure. I think that they would learn a lot by going back further than the last few years in their areas. They need to do directive searches. You don't just pick up a paper and read it. For the very specific things they are working on, it would be useful. For instance, I was reading Charlie Bachman's interview and I sent a note to my PhD student saying he's doing something that you are doing, you should reference it in your dissertation to give some historical context. The next step though from just digitizing is search and of course Google gives us full search through Google Scholar, as do others, like Microsoft. So we can use this and we can get access so much easier than when you and I were doing it fourteen years ago when we had to go to the library. If the library didn't have it we'd have to ask them to buy it, which would take weeks. Now it's available in seconds. So there is really no excuse for not having an accurate and complete provenance on the ideas that you're working on.

That brings me to a related question. So scientists complain that in Computer Science we don't cite things correctly. So for example, I've heard that every paper that we write about databases should be citing the original paper by Codd. We just don't do that. So should we? Or are they just talking from their own perspective?

I don't know how useful it is for everyone to cite Codd. I think that it is very important to put each person's work in context, and that's bigger than the last two years. But I think a more insightful citation, where you say "these are the important precursor ideas" and these are the best places where each of those ideas is described, I think would be the most efficient. So I guess I disagree with the scientists.

It also reminds me of the idea that there's nothing new under the sun. In fact, when Codd proposed relational databases it was actually the third time they've been proposed. The first I think being von Neumann in 1945 or something. But maybe our ability to ignore history and the fact that it failed all these previous times

[4] Recall that this interview was recorded in 2012.

[5] SIGMOD was held in Philadelphia one year later, in 1999.

[6] For more information of the SIGMOD Anthology, please go to `http://www.sigmod.org/publications/anthology/`

enables us to keep trying the same things and then finally the third time it would work…

I'm not sure they're the same things. They're probably the core of the idea, the kernel of the idea, but it's how you place it in the current context which determines whether or not it is going to be accepted. So, I had a colleague who said "I never read any of the research because I don't want to be biased. I want to do my own thing". I thought, how inefficient can that be? Because you're re-inventing the wheel that other people have already invented. So I think one can go too far and just spending all of one's time reading and one can go too far and not read anything. It requires some real skill to figure out where that middle ground is.

So you had 6 undergrads involved in your research projects last year under funding from NSF's Research Experiences for Undergraduates program. Why bother to write those grant proposals?

So number one, they're really easy to get. They fund almost all of them because they don't get very many. So if you want a few extra undergraduates, go for it. I think that six was too many for me. I think three would have been just about right. So there are a few tricks. It's important to use your graduate students to help direct them, but I just love working with undergraduates.

Why?

Because they're just starting out research, they don't know what research is. So it's this big, scary, wondrous world. They're not jaded at all, like some graduate students are and a lot of professors are. To them, it's totally new. They really are the future. Also I have to explain things very simply to them. It's hard to explain things simply, but when I understand it, I can, better. So that forces me to do that. Also, I'm a professor and this is my profession. Teaching is a part of it and so I want to find the brightest students and spend time with them.

Maybe I should add that my skeptical question is a little misleading, because we have huge numbers of REU7 students at my own group. Maybe it was a little misleading the way I asked that. I don't want people to get the wrong impression. I think we had five last year.

And they're wonderful to work with, aren't they?

We found it benefits both them and us. In fact, in my research center in Singapore, we've had forty-four interns so far and we've just finished our third year.

Wow, undergraduates?

Some are grads, but let's see…90% are undergrads.

That's incredible, that's wonderful.

It works really well for us. Okay, but this is about you, not about me. So let me ask my next question, which is that, I'm told that at work your door always open. How do you maintain your focus if your doors are always open?

I don't know if I do. I find it very hard to context switch. So I've been working at strategies for doing that. But I find that at the end of the day, the time I spend with my students is the most fun… much more fun than sitting by myself writing a paper, but I think it can get me very scattered and that's another trade-off that I'm still working on.

> ***I know how to make that rabbit run faster, but I don't know how to study that rabbit.***

But it sounds like that since you have the open door policy, in some sense you're benefiting more from the interruptions than…

Actually my door is closed, but I have a policy that you can knock anytime. So if they knock and I'm in a meeting, I'd say, "can I talk to you in a little while?" I find that if the door is actually just open, people walking by is very distracting. I work very hard to manage my physical space for more effectiveness.

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

I have no words of advice for practitioners because I am not one. I have great respect for them; they have their own set of challenges. For mid-career, one word of advice would be to figure out what you are best at (this is related to branding). And to really think about that and to do things that utilize that. Don Knuth once said that he picks problems for which he is the best person in the world to solve those problems, given his background. I think that's a great approach. So you have to really think deeply about what special abilities

[7] Research Experiences for Undergraduate (REU) is an NSF program

you bring, and I think that will also increase the passion, which is really important.

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

I do! It's the last book I wrote, on temporal databases. It really encapsulated the coordination framework that I've been developing over the last 20 years. It's three sets of three, and I like that symmetry. So it's different kinds of "time". So there are periods, instances, and intervals. There are three kinds of time in databases: there is valid time, transaction time and bi-temporal. And there are three kinds of queries: current, sequenced, and non-sequenced. All of that kind of came together in the book. It was really satisfying to see those ideas coming out in the new standard, which came out on October 2011.

If you magically had extra time to do an additional thing at work that you're not doing now, what would it be?

So I don't like that question. Let me tell you why specifically I don't like that question. Because it sounds like "what should I have done". You didn't say "should", but my philosophy is that (being a temporal database guy) the only thing that exists is the present.

No, but this is about your future!

That's right. So that's how I like to think of it. What would I do now that I did not do before?

That's right, if you had extra time.

Well, it's not if I have extra time because I'm not going to have extra time. It's what would I emphasize now versus not emphasize in something else. I really want to push ergalics. That's what I'm really focusing on. I'm not doing other things so I can do that.

Okay, if you can change one thing about yourself as a computer science researcher, what would it be?

For the future, what would I do differently as a computer science researcher? I need to learn a lot more about statistics and about the philosophy of science, because I'm not trained in that. I have an undergraduate degree in physics, but after that I was trained to be a computer scientist. So I know how to make that rabbit run faster, but I don't know how to study that rabbit. So that's what I've been focusing on. It's been really fun.

Great. Thanks so much for talking to me today, Rick.

Thank you.