Tamer Özsu Speaks Out

On journals, conferences, encyclopedias and technology

by Marianne Winslett and Vanessa Braganholo



Tamer Özsu
http://cs.uwaterloo.ca/~tozsu/

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 Providence, site of the 2009 SIGMOD and PODS conference. I have here with me Tamer Özsu, who is a Professor of Computer Science and a University Research Chair at the University of Waterloo in Canada. He is also the Director of the David Cheriton School of Computer Science there[1]*. Before joining the University of Waterloo, Tamer was at the University of Alberta, also in Canada. Tamer's research interests lie in multimedia databases, distributed data management and XML. Tamer received the SIGMOD Contributions Award in 2006, and he is an ACM Fellow and a former member of the VLDB Endowment board, as well as a former chair of ACM SIGMOD. Tamer's PhD is from The Ohio State University.*

[1] His term ended in July 2010.

So Tamer, welcome!

Thank you.

Tamer, you worked on object-oriented databases for several years and they never really "caught on". Do you think there will be a resurgence of interest in object-oriented databases?

Well, object-oriented databases never caught on as a replacement for relational systems. But a lot of the technology that was developed within the context of object-oriented database research made its way into object-relational systems. So, it is all there, but in a different context. The pure object systems do not exist. But object-orientation has actually made a comeback on the modeling tools, so there is a lot enterprise modeling tools, etc. that are object-oriented. I have recently listened to a talk by industry folks where the description was on the tools that people are now developing to take object-oriented models that have been developed and map them into relational systems. These tools are exceptionally complicated, and the hoops that you have to jump through are incredible, so if we had actually done object-orientation properly, then we probably wouldn't have had to jump through these. So it has never died, they exist in various forms, and in the modeling tools, they seem to be very popular and used today.

What do you think about this conference versus journal debate?

Hot topic. I am on the record as actually having stated that we have abused the conference system tremendously. I suspect the number of people who have read it are very small, but one of my SIGMOD chair notes in SIGMOD Record (I think it was in 2003) was on journals versus conferences. Perhaps the most relevant phrase I used there was that "we have done a wonderful job in convincing tenure and promotion committees of the value of conferences, but now we have to convince ourselves of the value of journals". So I think all of our attempts to "fix" the conferences, increasing paper submissions, avoiding bad reviews, unreliable reviews, variability in the reviews, etc. are really the outcome of pushing the system far beyond what they were designed to do. We have ignored the journals. That is actually our fault. We should have never ignored the journals to the extent that we have. A lot of the arguments that are made against journals no longer hold: long review cycles, etc. The top two journals in data management, which are VLDB Journal and TODS, have first round review cycles that are now shorter than conference review times, from submission to appearance. Reviews are probably more solid; they are more detailed, there is a cycle where you can respond to reviews and so on. I think we should revisit the value of journals.

So suppose we take our journals very seriously, what will our conferences look like then?

Well, I think one of the things is that conferences were supposed to be places for early dissemination of very new ideas. Right now there are far from that. A large chunk of the papers are basically incremental work that are very valuable, but they could just as well go into archival journals. The conferences should have their paper length reduced to about 8 pages, and they should actually focus on new ideas. The conference papers don't have to have all the i's dotted and t's crossed. They don't all have to have experiments. We should actually turn the conference to what they were supposed to be.

There is no reason that we need to have this many conferences either. We could actually have a few ones where the major ideas are presented, and the deeper technical content of the development of those ideas could lead to journal work.

But then, assuming that there is a certain amount of work trying to get out and be seen, how would the journals cope with the enormous increase in submissions?

Well, I think if you really compare it, there are a number of things going on there. If you compare the number of conference and journal submissions in computer science to some more established disciplines, there is no question that our numbers are far below what, for example, the chemists or the physicists or even some of the engineering disciplines produce. So, they manage to process it all within the context of a journal culture. The typical review cycles that are given in some of these journals in other sciences is about two to three weeks. You need to actually do the review and return it, now granted that the papers are shorter. But if we really change the culture, then our journal papers do not have to have 30% more content over already long conference papers to get to a size where it is really arduous to do the review. We can actually review them much faster. Now a lot of the journals are online, you can actually change it from print version to an on-demand print, so the page limit and the cost associated with page limits are no longer an issue. So there are ways of addressing this, ways that other disciplines have actually figured out, so that we can actually figure it out. We don't have to be identical to the other disciplines, we need to find our own culture, but clearly over-relying on conferences as the major outlet and the final paper publication outlet has stressed our system tremendously.

If we had actually done object-orientation properly, then we probably would not have had to jump through these [object-relational mappings].

So what is PVLDB, and how does it fit into this discussion?

PVLDB, to the extent that I understand it, it is an initiative of the VLDB Endowment Board led by Jag (H. V. Jagadish). Since I am no longer on VLDB Endowment Board, I don't know all the internal details, but to the extent that I know, at the steady state, the model that they wish to have is that paper submissions will be done to PVLDB. It is the Proceedings of the VLDB, which is an online journal, and is also included in ACM digital library. At the steady state, what will happen as I understand it, is that you will actually submit papers to PVLDB. And they guarantee a first round turnaround time of 1 month, so it is going to be very quick. But you will have the typical journal review process, which is you get the review, if they ask for revisions, you can do the revisions, respond to reviewers, comments, argue with the reviewers if you wish, and then you go through that more elaborate and probably better system of peer review. And then the papers will be accepted, they will appear as journal publications, in PVLDB, and the VLDB conference

program committee will select some of these papers for presentation. So if you have a paper published in PVLDB, you will get a journal publication, plus you make a presentation at the VLDB conference. That is the steady state that they are moving towards. Going from here to there is going to be tricky.

Right now[2] what is actually happening is that you submit papers the PVLDB, or you submit papers to the VLDB conference normally. There are no conference proceedings, the conference proceedings appear as PVLDB issues. And a certain number of papers that appear in PVLDB that have not been submitted to the conference directly, are selected by the conference program committee for presentation as well. But this is a transition as I understand, talking to Jag. The steady state will be that the journal will be the base, and the conference will be basically presentation overlay on top of that base.

So how does PVLDB relate to the VLDB Journal?

They are separate. I think, as I understand it, the VLDB Endowment is treating the VLDB Journal as a regular, classical journal. So you could have published papers at conferences before, you can increase [their content], the sizes are longer in the VLDB Journal than in PVLDB, and it has a regular [paper submission] track. What the relationship is going to be over the long run between PVLDB and VLDB Journal I suppose remains to be worked out. I don't exactly know what the VLDB Endowment Board discussions were on that one.

We have done a wonderful job in convincing tenure and promotion committees of the value of conferences, but now we have to convince ourselves of the value of journals.

So I see a lot of encyclopedias springing up, and I see that you've written chapters for some of them, I've written chapters for some of them… who is going to be reading all of these encyclopedias?

Well, I have not only written chapters for some of them, I am actually with Ling Liu, editing a major Encyclopedia in database systems[3]. But other than the one that we're actually working on, all of the others are really handbooks. So you have chapters devoted to them. Ours is somewhat different in that it is really a reference encyclopedia. There are regular entries, which are limited in the number of words, and we have definitional entries which are even shorter, some on the order of a page or so. So we really wanted to get a reference encyclopedia, more along the lines of Encyclopedia

[2] This refers to the state as of June 2009.
[3] The encyclopedia was published in 2009: Encyclopedia of Database Systems, Springer, 2009, ISBN 978-0-387-35544-3.

Britannica and so on, which are not really lengthy chapters, but shorter encyclopedia entries, with cross references to other entries. And the idea is that you never actually read an encyclopedia from cover to cover. There are some of us who actually like reading encyclopedias, but generally, you don't, you refer to them. So they are reference works. That is the context in which we operated in this forthcoming encyclopedia of database systems. It has about 1300 entries, will be about 5,000 pages published in 5 volumes. But, more importantly, it will be published online, so you can access it online and it will also be indexed in Michael Ley's DBLP, so you will be able to quickly get to the articles. So, it is a reference work.

So, who is publishing it?

It is Springer. Springer is publishing it.

So I'll pay to read it.

Well, if your university is a subscriber to the Springer link, and almost all North American Universities that I know of are subscribers, you get it free. All the authors who contributed entries, and there are about 880 them, get free online access.

So how long is the article on say, query optimization, you say they are length limited?

They are all length limited. I think the regular articles are 3,000 to 4,000 words, and the definitional entries are about 1000 words, so they are really relatively short.

Can you talk about the tradeoffs between stuffing XML data into a relational DBMS versus building a native XML DBMS?

Well in many of these, there are good arguments to make it either way. I mean, we have invested about 30 years of research into relational systems, they are very mature. You can cover a lot of distance by using the relational engines to support these complex data types, and so on. So there are arguments, but I've always (this goes back to my object oriented work as well) preferred to work in the pure object and the pure XML mode, just because whatever you do, some of those techniques will find their way into the other one as well. And I don't necessarily think that we need to actually have a one pony game, where we tie everything into the relational engine, and you do everything. It almost reminds me of a Turing machine. We know that anything we can compute, we can compute with a Turing machine, but none of us are actually talking about programming Turing machines.

Well, maybe we are, because of virtualization!

Well, I mean, computer science is all about abstractions, building abstractions, and virtualization is an abstraction that hides certain things that you do underneath. But the issue is that there are parts of the relational engine, in the relational technology, that are definitely relevant in the XML world, or any other world. But that doesn't necessarily mean that you basically tie everything to the relational [technology] and figure out how to map a complex data type to a flat tabular structure and do the processing there. I think we need to separate the technology that we

developed for relational systems, from the relational systems themselves. There is room for the technology obviously to play a role in declarative querying, optimization techniques, etc, and even some of the optimization techniques that we use can certainly carry over to how we support these more complex data types. But we don't need to actually force everything into the relational engine. I think that probably is not the right thing to do. Even the most recent discussion of column store versus row store, is an indication that not all applications need to be forced into a single architecture for data management. I think that we need to be able to break out and figure out what are the really critical essentials that we build into very [small], perhaps micro-kernel type engines, and what needs to be left out for customization, for different types of applications, and different types of data types.

Interesting. So have you built a system like that?

Well, we started building one in the object world, and we actually went quite a bit, and then we got distracted with other research interests. We started in the XML world as well, but by the time we were doing the XML world, I really did not have the energy to build up a big implementation group anymore. I did that in the '90's and it was really a lot of fun, but I didn't want to repeat that one more time. So we really never pushed it to a reasonable prototype on the object side. We did have internal prototypes that we actually fooled around with for research purposes. But we did a little bit of it.

So benchmarks: what do we need for XML?

Well I think there are benchmarks that have actually been developed. We did one benchmark for XML called XBench, that started in collaboration with IBM Toronto Labs, and then we kind of took it on our own and went further. The fundamental point of our benchmark was that we did not actually want to just say, well, "what are some interesting queries, and what are some interesting XML structures that we should test these systems on", but we tried to go out and actually find actual customer data, and IBM helped quite a bit. They looked at their customer data, we didn't see the data, but at least the characterization. "What is going on in terms of the types of XML data that people develop, and what type of applications are we seeing"? So we came up with a taxonomy of the types of XML data that we were seeing, and then we developed a family of benchmarks that really had their roots on the statistical characteristics of the data that we were seeing in real life, for the most part. There were parts of the taxonomy for which we could not find the data, for which we went and looked at XML use cases, and other things. I think the important thing in the benchmarks is basically being able to defend the choices in the benchmark that you are making.

… not all applications need to be forced into a single architecture for data management.

So did you get a query workload too?

Yes, but our query workload at the time when we did this, which was very early 2000's, there weren't actually that many applications being built, so the query workloads are really a distillation of, a careful analysis of XML use cases that were reported as part of the XML standardization efforts. We looked at it and said "okay, what classes of workloads are these use cases representing", and work from that.

One of your colleagues have asked me to ask you when the 3rd edition of your textbook Principles of Distributed Database Systems will be out, and why is it taking so long.

A fair question and a touchy question! It is going to be out soon, how soon, I don't know[4]. The third edition is a major rewrite, and includes substantial new material. Every chapter has been reworked, but also there is a lot of new material that we didn't fit into previous chapters: replication, peer to peer systems, work data management, data integration, in a much fuller sense than we had before, etc. We have one chapter to write and two chapters to revise and then we'll be done. So, soon! The reason it took this long is because I took over the directorship of the school, and it is just time.

I understand that you used to be active in politics.

Yes, I was, when I was at university, and shortly after that.

And where were you?

That was in Turkey. I was a fairly left wing radical, more interested in that than in school things.

So what made you become more interested in CS?

Well, I think I was always interested in the CS part. During my undergrad degree, and actually my first master's in industrial engineering, I was always interested in CS. I kept taking courses in CS. Even when my degree advisor told me I could not take more CS courses, I kept taking them, so I graduated with extra [credits]. I was the type of student who did very well on the topics that I liked, and just barely survived on the topics that I didn't care about. So I was always interested in CS, but the specific interest in databases goes back to about 1976, when I was doing a part time masters, and working at the Turkish postal administration on a problem which basically can be the directory problem, the 411 system. You call up and you actually ask for a number, and we were supposed to be designing a system, but the system needed to be able to be queried using different keys, and we were struggling on how, what data structures to use, and how to lay it out so you could actually query it multiple ways. I was part time taking a graduate course, and part of the course was databases. The first book, Date's first edition, had just come out. We had a visiting professor from US who was giving a course, and light bulbs when on. I said "that is the solution to the problem that we were doing", and I got hooked up on databases. Back in '76.

So is that the approach you actually used your directory problem?

[4] The book came out on March, 2011.

Well, we didn't because there were no [database] systems then. I mean we used a general approach, but we couldn't use the system. You still had the index sequential stuff, so you had to build multiple indexes over the data, etc. But the general idea of how to lay it out so you could actually do it was it. Shortly thereafter, I started working for a United Nations project in Turkey, and I remember going to my first conference. It was the second VLDB conference in 1976 in Brussels… That's how long ago I got involved in databases.

How many students of yours have picked up your hobby of collecting pens?

I'm sorry to say, none. Although, we gave a fountain pen as a gift to one of my former students recently, so we are working on them, but so far, as far as I know, none of them have picked it up.

Well then, if they don't like it, why do you like it, why do you collect pens?

Well, I actually don't know why I like it. I think there is tremendous esthetics in fountain pens. They are very simple devices on the face of it. But they are really very sophisticated. Somebody actually wrote a whole monograph on the physics of fountain pens. The capillary action, why, what happens, how the air goes in and the ink replaces air, etc. You could get hooked on them for scientific reasons. Mine wasn't for scientific reasons, I just loved the esthetics of it, and I loved the feel of it when I write. I don't always use a computer to take notes. I use the old pen and paper approach. It has grown from an interest to a sickness.

You are going to be in this business for 30 plus years, so if you don't enjoy it, don't do it.

Oh, a sickness? How many do you have?

I think right now I have about 600.

That's a lot of pens. Do you insure your collection?

No, I don't. And I have some of them on my webpage. And my wife keeps asking me when we should expect somebody to see the webpage and then storm our house and steel them.

Actually, I didn't see them on your webpage. On your homepage I saw photos of your wife, son, dog, and your motorcycle, no pens.

Actually, if you go one below that, there is a link to my pens, so you missed it. It is there.

So why does that motorcycle rate above the pens?

This actually came from a talk I was preparing, and they said "we want you to talk about what you do outside of work". So I said, well, I spend time with my family, I walk my dog, these are things I enjoy, I collect pens, I had the link, and I love to ride my motorcycle, so I actually added them there so they appear. So the motorcycle is another passion.

Do you have an interesting story about one of your pens?

Well, every pen has an interesting story…

Oh, just one! Just one!

I'm going to have to think about it. Either the pen is interesting, or the purchase is interesting. One of the interesting stories is we were in a taxicab in Taipei in 1995 Data Engineering with my student, coming back from a museum. We were on a bridge, and I saw the glimpse of a pen shop, a hundred meters down at the other end of it, and we stopped the taxicab and went down and I bought the pen. My student could not believe that I could actually spot the pen shop that far away, about 50 meters below the bridge, a hundred meters on the other side. But us pen collectors have an eye for these things.

I guess so! Do you have any words of advice for fledgling or midcareer database researchers?

The word of advice is good luck. It is really far more stressful than when we started. Our expectations for hiring and tenure have grown to be what one might actually legitimately call ridiculous stage right now. If you take the acceptance rates and you look at what we expect, a PhD needs to start publishing a paper, or at least submitting papers, in their second term, which really makes you question whether that is possible. But you know, the pendulum swings in this business, and it will. I think a fundamental issue is, do what really attracts you, I think that is probably very cliché, but you are going to be in this business for 30 plus years, and if you don't enjoy it, don't do it.

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

Well, that is probably tough to say, it is almost like choosing which of your children are your favorites, so it is probably quite difficult. But I think the work that we did, even though we talked about it not making it in the main stream, the work that we did on object orientation was the most enjoyable because it really spanned fairly theoretical to system type of work to fairly pragmatic work, where we built query optimizers and tested them, etc. That and the multimedia image database work that I did in the second half of the 90's were end-to-end projects that we actually started from the architecture and the models all the way to languages and implementation. So those were probably quite interesting projects.

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

Magically finding time is probably the right word. I'd love to do a lot more reading, which I'm not able to do right now.

I think you're supposed to say that you will finish your 3rd edition!

I'm trying not to think of that one! But that will get done in the next 6 months.

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

One thing that I wish I was better at was on more formal aspects. I can actually do certain things, but I'm not that good in the theoretical side of databases. And there are lots of interesting problems that I can skirt, but I cannot dig in there. So I wish I were actually a better theoretician to be able to tackle those.

Thank you very much for talking with me today.

Thank you for the opportunity.