Serge Abiteboul Speaks Out

on Building a Research Group in Europe, How He Got Involved in a Startup, Why Systems Papers Shouldn't Have to Include Measurements, the Value of Object Databases, and More

by Marianne Winslett

http://www-rocq.inria.fr/~abitebou/

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 SIGMOD 2006 conference in Chicago, Illinois. I have here with me Serge Abiteboul, who is a senior researcher at INRIA and the manager of the Gemo Database Group. Serge's research interests are in databases, web data, and database theory. He received the SIGMOD Innovation Award in 1998, and he is a cofounder of Xyleme, a company that provides XML-based content management. His PhD is from the University of Southern California. So, Serge, welcome!

Serge, you have built what may be the most successful database group in Europe. How did you do it? And are the challenges for building a group different in Europe than they would be in the US?

Many people deserve credit for the group's success. It is a continuation of a research group named Verso that was sponsored by Francois Bancilhon and Michel Scholl; so when I arrived, the group was already existing. My contribution was to bring in some new fresh very good scientists --- Luc Segoufin, Ioana Manolescu, Sophie Cluet. More recently, we moved to a new research unit of INRIA. The idea was to get closer to the university, so now we are at the University of Orsay; we merged with the knowledge representation group that was managed by Christina Rousset. Besides getting closer to the university, the idea was also to bring together specialists in database systems and people from knowledge representation, because we think this is what is needed to really attack the problems of the web.

Based on that experience, do you have any recommendations for other people who are trying to build strong groups in Europe?

It is very simple. It is just like building big groups in the US. You have to bring together talents, you have to shoot only for the best people and try to convince them to come to your group. That is not always easy, but that is what you should do.

How is database research different in Europe than in the US? As a European database researcher, do you ever feel left out because you are not living in North America?

Database research got a late start in Europe. I did my PhD in the US, and when I came back to Europe, there were very few groups doing databases. Basically, the interesting work was going on in the US, so we had to catch up. I think to a certain extent, Europe has caught up now.

Another difference is that the main database companies are in the US, so to do something with industry is not very easy in Europe. On the other hand, now that we have built a strong database community in Europe, the fact that the big companies are not present is perhaps an advantage. I believe in the future much of the interesting research is going to be about web data and probably be driven by small company startups, and those we have in Europe.

Recently we have seen several well-known US database researchers move back to their home countries in Europe---people like Peter Buneman, Timos Sellis, and Yannis Ioannidis. Is this a trend, and if so what do you attribute it to?

I think it is clearly a trend, and one that I love, personally. What's the cause of it? The database research in Europe now is at a very reasonable level, so the funding is getting better. Maybe also the political situation in the US would explain it to a certain extent. The government you have now in the US is probably not attracting too many people.

Do you mean because of the Iraq war, or because of the lowered funding for database research, or because of everything?

I think because of everything, but primarily the politics.

Your research group is one of the few in the world where the majority of the members are female. How did that happen?

One reason is that Francois Bancilhon never made a distinction between women or men researchers; he always wanted the best people. He found very good women for the group, and I have been trying to continue this tradition. Since I became the manager of the group, we hired four permanent researchers. Two are women, and two are men. I didn't do it on purpose, I just chose the ones I thought were deserving of the jobs. I mean, I don't choose alone, but this was the result.

One of your colleagues told me that you are "very feminist, but in a French way." What does that mean?

Thank you to my friends! "Very feminist"---I believe that that is something that I am. I have always considered that men and women should be given equal opportunities. In my career, I have had the chance to work with women like Jennifer Widom, Sophie Cluet, Tova Milo; that only reinforced this strong feeling. Now, "feminist, in the French way", what does that mean? Maybe the difference with the American feminist is that we don't try to believe that women and men are just the same. We do see differences, and we like the difference, but we try also to encourage equality of opportunity between the genders.

There is still a long way to go. Just as an example, I was recently in a workshop organized by Microsoft, called "Towards 20/20 Science"; which was supposed to set up the agenda for computer science to help scientists at large. There was a huge table with physicists, chemists, biologists---every kind of science was represented. And we realized that around the table, there were mostly men. That is ridiculous; we can do much better than that. There are tons of great women scientists, and we should pay more attention to the equality of gender.

Would you recommend that new computer science PhD graduates in the US consider a job in Europe?

Absolutely! I think it is still easier to get a position in a good European university than in a good American one. I think that going to Europe is something that new graduates should really consider. Also, Europe is nice, so that should give them a great experience.

You started out as a database theoretician, but have moved more and more towards the practical side. Recently you even got involved with a startup. How did that happen? I'd like to hear about the interplay of the theoretician and the practitioner in you.

This is a very long story with many aspects. I will tell one part of it: how did it start? I was at Stanford at the time, visiting for a couple of years. My friend Francois Bancilhon was interested in what we were doing at Stanford. I was explaining to him about semistructured data and since he is in industry, he said, what good is that going to do for industry? So we started to discuss it, and then we had regular telephone meetings with the idea to start a company. I took it as a challenge. I believed that semistructured data could be very useful, and I had to prove it, at least to him.

Then we brought Sophie Cluet on board. We worked on the topic for a year and were developing the software that became the Xyleme system, in parallel with these business oriented discussions.

The interplay of the startup project with research is interesting. The startup grew out of theoretical research that I had done before, and when we started the company, I thought this would be a dead time for research. And it turned out this was not at all the case.

Getting involved in Xyleme actually brought a lot of inspiration for new research problems, some of which I am still working on now.

What research problems did you find that you hadn't thought about before?

In the early days, we were developing a page rank algorithm for a web search engine. That was a lot of fun, but it required a lot of resources that we did not have. Google can afford to have tons of machines to store the graph of the web, but we could not. I thought that there had to be a way to do the ranking without so many resources. With some students, we developed an online agorithm that computes page ranks without having to store the graph of the web. Then there was the analysis of the algorithm, and we started working on the question of what happens when the graph evolves; so there were lots of open questions.

When you describe the company, I don't see directly the connection to XML-based content management. So what is XML-related in that particular problem?

To understand, you have to go back to that time, when XML was just beginning and we had the crazy idea that XML would conquer the world. We thought that five years later, everybody would be publishing XML on the web, and we were going to provide a query engine for the entire XML of the web. So our goal was to be able to find, index, and query billions of XML pages.

Of course we were wrong. But then we realized that even if the web did not have so much XML, inside companies they do have tons of XML. We changed the business model and now we have aproduct that can find, index, and query all the XML in a company and enrich its content using semantic tagging, linguistic analysis, and so on. The product scales very well because we intended it originally for all the XML of the web, and a company usually does not have so much XML, so the product goes very fast even with very large volumes of data.

And does the page rank algorithm come into the picture in some way?

The page rank was abandoned; it was just a nice research problem.

Why didn't you move permanently to industry?

In small companies, like Xyleme, the beginning is great from an engineering viewpoint, because you are doing a product, you are doing a system, and that is lots of fun. You are meeting customers, and that is great. But after a while, it gets boring. You have all these good ideas for improvements to the product, but the managers tell you it is going to be too expensive to do them or---the worst response---that the customers don't want that. How could the customers want that improvement if they don't know about it? And if you refrain from doing anything new, it becomes kind of boring after a while.

Also, from a customer viewpoint, essentially you are trying to repeat the same sales again and again, selling the same thing, which is the opposite of research. In research, once you have done something, you don't want to do it again. So my experience is that in a startup, or in a small company, the interesting part is the marketing, the business part; the engineering part soon becomes boring.

Do you see differences between the database theory and systems communities, for example, in the way program committees function or the way works are evaluated?

Yes, I think there are big differences. It is very easy, to a certain extent, to evaluate theory. You look at it, you see whether the definitions are elegant, you see whether the proofs are deep. You can measure it.

When I started to work on systems, I thought I was going to learn a new culture, which is very interesting. But in a way I was disappointed by the way systems research is evaluated. It is much more difficult to evaluate a system than it is to evaluate a theorem. People are supposed to present performance measures, but my experience is that when you look seriously at the experiments, most of the time the results are kind of trivial: what you find is what you would expect, and it is very difficult to compare different approaches. My take on it is that there is a lot of randomness in systems program committees, much more than in theory program committees, and I don't see any way to improve that.

There is almost a law in system conferences that you cannot publish a paper if you don't have performance measurements. I think this is dumb, because most of the time the measures you see are really some vague experiments that were put together by students in a couple of months and that don't teach you much. I think measures are important, but to produce real measures takes more than a couple of months and a couple of PhD students. So I would rather see some system papers evaluated based on the ideas and functionalities they propose, and leave performance measurements to those papers that are really talking about optimization and performance issues.

I claim that all good systems ideas are shallow. The flip side of that is that if an idea is deep or complex, then it's probably not going to work out when you build it. That dichotomy might have an impact on the evaluation process too.

Do you think the page rank idea at Google was shallow?

Sure, it is a great simple shallow idea. If you can't present the idea in, say, two sentences, then it's never going to fly if you are really going to have to build the system.

I have immense respect for the page rank idea, because everybody *could* have had it.

Yes, that's the hallmark of the best systems ideas! It's shallow, and anybody could have had it. In hindsight, it looks so obvious, but nobody had done it. [As another example: "let's keep all our data in tables."]

But you have to think about the idea first, and you have to believe it is going to work, and you have to make it work. That is what a good system idea is. It must be simple, but then you have to prove that it works. I think this requires engineering and good ideas, and belief.

But in the systems papers, if you don't build it, then how can you argue that it works? And then if you build it, you can measure it as a way of showing that you've built it and it works.

I'm more into prototyping: you do a system to show that it can be done, that it has reasonably decent performance, that all the functionalities are present, that you didn't miss any important point. Requiring that besides having this great idea and making it work, you have to also show performance measures---this is ridiculous, because time spent measuring a prototype is time not available for adding functionalities. I am more interested in functionalities and proof of feasibility. Only after that will I be interested in performance. Of course, if what you are studying is XML query optimization, then it doesn't make sense to have a paper without measures. But if you have a novel idea, I think proof of feasibility is enough.

Sometimes measurements are made because the reader will want to know what price they will have to pay for your great new functionality. For example, they might have to give up 10% in performance if they adopt your new technique instead of doing things the old way. So you can show how good your idea is by showing that you don't have to give up very much performance in return for getting the new functionality.

Sure, things are like this sometimes, but providing measurements shouldn't be a *law*.

You have worked a lot in areas that were unpopular or controversial at the time, such as nested relations, object databases, and semi-structured data. How do you choose your new research topics?

My taste is always to go for the new things. I like a topic when it is fresh and new. Perhaps this is because I am lazy: if you go into a new research topic, you don't have a zillion papers to read. Of course, if everybody preferred to work on new topics, it would be a nightmare, and I would have to choose a different approach.

Ultimately, I choose a research topic based on the people I want to work with. I choose a research topic because I am going to have fun with it. So fun and pleasure are prime criteria.

Mike Stonebraker refers to object databases as "a zero billion dollar market." Does this mean that the research community shouldn't have worked on them?

There is a big contradiction between the two sentences. Mike Stonebraker knows the industry much better than me, and the statement is about industry; it has nothing to do

with *science*. Mike's statement about zero billion dollars is absolutely no argument at all against the scientific value of research on object databases.

Actually, I might even disagree that object databases are a total failure in the marketplace. My wife, Sophie Gamerman, was a VP at O2 Technology, and we did make some money out of O2 Technology in the family. So I am really thankful for the object database industry!

Now, from the point of view of research, I think object databases have brought a lot of very good ideas that have strongly influenced the field. For instance, look at the XML world. With persistent XML, often you are playing with the Document Object Model (DOM) interface. To me, DOM is an object repository. So when you are doing persistent DOM, you are just doing object databases, whether you like it or not.

The funny part is that you shouldn't say that you are doing object databases. Some venture capitalists asked me to do a technical due diligence review for a startup. After half an hour of listening to the startup's founders, I told them that what they were doing had already been done by the Object Data Management Group (ODMG, www.odmg.org), and asked them whether they were aware of it. They told me that they did know that they were doing object databases, but that they didn't want to mention it because that was not a good way to raise money. These people were doing object databases, they knew about object database technology, but they didn't want to mention object databases because people have been going around for ages saying that object databases are bad technology. Object databases are not a successful industry, but they are a very very successful *technology*.

How does tenure work in France?

It is very different from the US. Once you finish your PhD, typically you have to do one or two years of postdoc. Then you get a permanent job, either in the university or in a research institute such as INRIA. But you don't have really a tenure system.

Is that good?

I don't think it is good. I think it is a bit too early to see whether the person really likes research, and is really good at research. The pre-tenure time in the US may be a lot of stress for the people undergoing it; but on the other hand, when you tenure somebody after five or six years, then you know that the person is built for research.

Someone suggested that I ask you whether you think Xquery stinks. Do you want to comment on that?

I know it is very popular now to do Xquery and XML schema bashing---we have heard some here at SIGMOD 06. I don't participate in it.

If I had designed Xquery, I would have done it differently. I would have made it more functional. I would have been perhaps further away from SQL. But I was not on the committee. The people who were in the committee put together a proposal, and it is a compromise of course, so it is not perfect. But at least we have some standard, and it is good to have one.

I think to a certain extent, focusing on Xquery is ignoring the real problem. Xquery lets you query a local repository of XML, which is not the real problem. XML was originally proposed as the data exchange language for the web, so what we need is a language that allows you to talk about distributed XML resources and distributed data resources in general, and query them. I have been trying to do that the last few years with Active XML with some colleagues --- Omar Benjelloun, Ioana Manolescu, Tova Milo, and others. It is good that some people work on XML repositories, and XML processing, but I think there should be more work on distributed query processing in the web context.

How does a researcher's character and personality affect their success as a scientist?

Research mostly involves working with people. Some people work alone, but most people work in groups. Your human qualities really affect the entire group. A group should produce more output than the sum of the work of each person separately, and to make that happen requires not only intellectual talent. It requires the talent to explain to the other people, to listen to what they are saying, to try to work together. That's not easy. Personally, I have the reputation of not being very easy to work with, but on the other hand, many of my coauthors became very good friends.

You recently published a novel, Sparrows on the Web *(http://sevres-pratique.com/Serge/). To what extent are the computer scientists in the book inspired by real-life characters or experiences?*

One of the facets in the book is a startup company that's developing a search engine, and of course I have been exposed to some characters like that. And of course, the characters of my books---I have written more than one---are often influenced by people I know. But you shouldn't look further than that, don't try to recognize anybody. There have been cases where people have tried to recognize characters in my books. Once I got an email from a lady who thought she was one of the characters, and I had never met her before. So don't try to recognize somebody in my books.

In addition to writing that book and another novel, you do sculpture as a hobby, have a strong interest in politics, and have a family at home. How do you make time for everything and everyone?

I am Superman. You shouldn't tell anybody, but that is what I am.

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

If you don't think that you can be productive as a researcher, then you should try to do something else. Develop systems, or go into management, do something easier.

Among all your past research, do you have a favorite piece of work? Was it also the most fun to work on?

Yes, I have one, it was some work I did a while ago with Victor Vianu. We were working at that time on fixed point logics, and we were really puzzled by the fact that there are certain very simple queries that you cannot do with first-order logic, with relational languages. We had been working on the topic for a while, and had written several papers. Then at one point we were at a blackboard, and we designed this notion of equivalence classes, and suddenly everything started being very clear. After that we got theorems (because in papers you always have to get theorems), and the theorem is something like "P-TIME is equal to P-SPACE if and only if fixed point logic is the same as partial fixed point logic." Nobody cares---well, some people seem to care---but for us, the great thing was the understanding of these equivalence classes. When we understood it, we thought it was beautiful. I really had a great time, and I think Victor shared that great time.

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

I don't want extra time. If I had extra time, I would write one more paper, review more papers, get more administrative tasks, so if I had to choose, I would rather have a little less time.

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

I never go deep enough into stuff. I like to write first drafts, but I hate polishing papers; I find it boring, but you have to do it. I have made a lot of effort to get better at it.

There is the lesson of my late friend Paris Kanellakis, who disappeared with his family about 10 years ago. I was working with Paris at that time on IQL, which is a formal model for object databases. He was forcing me to go over the model, again and again. I think we wrote the definition of the model perhaps 40 times on the blackboard. Each time it was just a little bit cleaner, just a little bit better notation and so on. At that time, I was getting irritated because I wanted to move further, to go faster. But when I look back at it, I really love this work, and I think I love it because it is very clean and the time we spent on it was really worth it. So what I would change about myself is that I would try to be a little bit more thorough in the work I do.

Thank you very much for talking with me today.

Thank you for inviting me.