Carlo Zaniolo Speaks Out on his Passion for Relational Databases and Logic

Marianne Winslett and Vanessa Braganholo

ScAi
SCALABLE ANALYTICS INSTITUTE
Mobilize Laboratory
Web Information Systems Laboratory
Intelligent Data Exploration and Analysis Laboratory

Carlo Zaniolo
http://web.cs.ucla.edu/~zaniolo/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are in Phoenix, site of the 2012 SIGMOD and PODS conference. I have here with me Carlo Zaniolo, who is the N.E. Friedmann Professor in Knowledge Science at UCLA. Before that, Carlo was a researcher at Bell Labs and MCC. His PhD is from UCLA. So, Carlo, welcome!

Thank you, it's a pleasure to be here.

So how did you get into the database field?

It's a long story. We'll have go way back to the glorious early days of the relational model, right? Particularly for me, the relational model was a real savior, and I'll tell you why. As soon as I was done with the PhD courses and preliminary field exams at UCLA, I had to leave to, believe it or not, to serve in the Italian army. I had obtained a study deferment, but that eventually ran out and I had to go back. But before leaving, I talked with my advisor, Prof. Michel Melkanoff, and told him, "I want to select a topic, because while I'm there, I will have a lot of time, and I can start my PhD research". And so my advisor gave me the latest works by E. F. Codd.

That was love at first sight: I read those four early papers by Codd and they were magnificent papers. I took them with me to Italy, and for two years I worked on the ideas in those papers all alone – without any external communication, no Internet back then – at times I might have talked to myself! When I returned to UCLA, I had already started on the theory of multivalued dependencies and related topics. Of course, this made a big impression on my advisor who told me something like "I gave you those papers, and I agreed you should try to work on them for your PhD thesis, but my expectations were very low". So he was very impressed and that was a great start. But at some point, without us knowing, people like Ron Fagin had started working on similar problems (i.e., how to go beyond Third Normal Form to cure problems caused by new dependencies).

I'm very pleased to see that Datalog is being rediscovered.

In fact, as soon as I finished and filed my PhD thesis, and my advisor sent it over to E. F. Codd, we immediately got back a letter from him saying: "Oh, I gave it a short glance. It looks very interesting. In fact there's a guy here (Ron Fagin) who is doing similar work". To show the independence of their work, since they didn't even have a typed report yet, E. F. Codd sent over the handwritten manuscript that Ron had prepared for the IBM typists. To me, that felt like a very bad surprise. But in retrospect that was also a blessing since everybody knew Ron Fagin and E. F. Codd, and the fact that I had gotten those results before them, and working in the isolation of an army barrack, showed that I could do as good research as them, working as a team in the Mecca of IBM DB research. Frankly, I surprised myself too, because my background and experience till then was that of an electrical engineer. In fact, I had worked through most of my PhD years as an electrical engineer and CAD programmer. But those early times were special, and everything was possible in those days. So the database relational model was my first love in Computer Science, and it basically stayed with me for the rest of my professional life.

You mentioned logic, okay? That was probably my second love, since I was very involved with Datalog. In terms of references, those papers on Datalog do not get a very high citation count because this is still a slow time for Datalog, But I'm very pleased to see that Datalog is being rediscovered. In particular, a number of results on non-monotonic reasoning that are now being rediscovered find applications in various situations. For instance, Joe Hellerstein and his UCB students are using non-deterministic choice models in their works, and they are generous in their references. There are also other pieces of work such as those on XY-stratification and monotonic aggregates which I'll probably tell you more about later.

Well since we're talking about logic, let me ask you about the mid-80s, when the Japanese launched the Fifth Generation Project. What was that about and how did the US respond?

Well, you probably know the background: for many years, the US automobile industry dominated the world; but then they started relying more on marketing than on improved engineering and manufacturing. So, the US car industry found themselves threatened by the Japanese industry that was providing reliable manufacturing, nice models, good mileages, etc., etc.

That was also the time in which Expert Systems came about with much hype; at that point, Japan announced the Fifth Generation Computer Project and Institute saying: "We're going to make expert systems at an industrial scale and they are going to define the next generation of computing". Immediately, people concluded that the US computer industry was going to experience as big a threat as the one the automobile industry was facing. So, there was a major computer industry initiative and a research consortium was founded with the participation of several US companies. It was named Microelectronics and Computer Technology Corp. (MCC), and to head it, they chose the four-star admiral Bobby Inman, who was the former head of the NSA. So that started off with a bang, okay?

From my vantage point, that was the perfect time to leave Bell Labs, which was experiencing a major crisis due the end of the consent decree. So I decided that, having learned that industry cannot offer stability, I should instead look for the best opportunity. Sure enough, opportunities were great at MCC, for a while. Then, stability became a major problem: some of our funding companies went out of business or they were restructured; also companies discovered that is hard to collaborate, and that the great results produced by research might not match the narrow needs of the companies that sponsored the research. So, after eight years of so, MCC was pretty much a thing of the past. Actually, they were a bit unlucky because soon after that, the Web revolution happened. If they had hung on until then, they could probably have been rescued by some new initiative or some big company that needed the technology. But at that time, expert systems had not delivered on the their inflated promises, and things were not particularly nice for database research either, with disputes between object-oriented DBs and deductive DBs, which were not productive at all.

I had kept in touch with universities, and UCLA offered me a prestigious chair in Knowledge Science, which I accepted, and I am happy I did. Believe it or not, before going to academia, I had worked in industry for 20 years. So, moving to academe involved a long adjustment process; but in the longer term, things seem to have worked out, a fact underscored by this best paper award[1].

This paper that got this great award might or might not be the best paper I ever wrote, but, certainly, it is not the worst, and in fact it is a great paper. To me, it's a comforting statement that I can still produce good work after so many years have passed. But I am not the only one at that: Bruce Lindsay[2] today didn't speak like someone who was retired, right? He spoke like a person full of ideas and energy. So, I guess we entered an era where people with gray hair can still make great contributions.

So what's the award paper at SIGMOD about?

This is another interesting story. Basically, there has been a sequence of interesting developments coming from the following simple idea (breakthroughs often come from simple ideas): why not use Kleene-* expressions (i.e. ReGex) to find recurring patterns in data streams, and in sequences. That line of work started with a paper by my student Reza Sadri et al. in 2001 when we extended SQL with Kleene-* constructs[3]. We also had some nice optimization methods based on extensions of the Knuth, Morris and Pratt algorithm. In fact, there has been some recent initiative seeking to extend the SQL standards with similar features[4].

So we (i.e., Barzan Mozafari, with Kai Zeng, and me) continued working in that area, and, as it happened, recent advances in formal languages and automata had just introduced the Nested Word model which is significantly more general than traditional regular expressions, since it handles parentheses, nested structures, etc. This model can be implemented very nicely through what is known as Visible Pushdown Automata, which are automata that do not have the complexity of those required by context-free languages. We first used these new techniques in SQL, to allow nested Kleene-* expressions and that was nice. But, perhaps, the area that needed this new technology the most was XML, for which ad–hoc language extensions had previously been proposed. To a large extent, what we have in our paper is similar to those, but along with language constructs we have now provided a technology which makes them amenable to efficient implementation. Thus we generalized XPath nicely, and also provided an optimized execution engine for that. To make things even better, this new technology supports not only XML, but also other kinds of nested expressions as well, including logs of program executions (with nested calls), or also some RNA sequence analysis, and temporal database queries–and there are still a number of unexplored opportunities in this area. We should be very grateful that the Nested-Word advances came along at the right time.

Yeah, it sounds very interesting. You mentioned something about the SQL Standard?

I will have to do more research on that. The last time I kept in touch with that, Oracle was very much pushing for that[4].

Pushing from which extension?

To provide the ability to specify regular expressions for searching ordered sequences and data streams in SQL.

[1] Carlo Zaniolo won the 2012 SIGMOD Best Paper Award for the paper with reference: Barzan Mozafari, Kai Zeng, Carlo Zaniolo: High-performance complex event processing over XML streams. SIGMOD Conference 2012: 253-264.

[2] Bruce Lindsay is the recipient of the 2012 SIGMOD Edgar F. Codd Innovations Award.

[3] Reza Sadri, Carlo Zaniolo, A.M. Zarkesh, J. Adibi: Optimization of Sequence Queries in Database Systems; PODS 2001.

[4] Fred Zemke, Andrew Witkowski, Mitch Cherniak, Latha Colby: Pattern matching in sequences of rows, Available at http://www.docfoc.com/pattern-matching-in-sequences-of-rows-march-2-2007-change-proposal-for-sql.

Okay. This week at the conference when I was talking to people about the work you've done, the paper that popped to their mind immediately as having the most influence was the 1983 GEM paper[5]. What was the new idea in that work?

The new idea in that work came from observing that the relational model had limitations. In particular, it did not support well the notion of "entities" (later called "objects") with hierarchies. Coming from a relational database background, and being very loyal to the relational model, I noticed that simple extensions, could fix that. So, we introduced into SQL constructs that support path expressions (to simplify the specification of most joins) and entity hierarchies. We did that in a rather simple way, by first formalizing the idea, and then proposing an implementation which did not have a dramatic performance overhead. So that was the right idea, which was proposed at the right time, and got much attention by remarkable people. In particular it was included in the series "Readings in Database Systems", edited by Mike Stonebreaker et al., several times.

So, that was a good piece of work, but it had somewhat strange longer-term consequences. Indeed, later on, I became a deductive database person; but by then, people would keep calling me about object-oriented problems and job opportunities. You know, I wanted to tell them: "Yeah, they are both good ideas, but don't push them too hard to the extreme, thinking that they will change the DB world completely. Be reasonable in what you can do with them". In fact, the kind of extensions that GEM was proposing eventually made into object-relational databases. Likewise, some of the recursion work (from Datalog) also made it into relational database systems. Moreover, I think that there is still some more untapped potential there (i.e. on recursion) with some techniques we have not exploited yet, and they might actually be applicable to the Big Data revolution that we're experiencing now.

What kind of things for Big Data?

Do you want me to tell you what my next paper is going to be about? I'm not sure because I haven't written it yet, but, of course, the first thing that comes to my mind is graphs: there is a lot of interest in graphs. You know, the recursion of Datalog is a natural for graph applications. But we still face serious challenges there: as you know, in recursion we are restricted to monotonic operators and therefore we cannot use arbitrary aggregates. However, it turns out that certain kinds of aggregates can be used and that allows us to express a large set of new algorithms, which, before, were not expressible or were expressible in very efficient ways. Therefore, we can now reduce those graph problems to the standard framework of recursive queries that are implemented using semi-naïve fixpoint, magic-sets and techniques like that. As it turns out, there has already been work showing that recursion can be implemented efficiently in MapReduce, and thus we can build on that to support efficiently a much wider range of applications in Datalog: we can express things such as Markov Chains using standard recursion.

I guess we entered an era where people with gray hair can still make great contributions.

Let me go back to the time at MCC. I think it must have been a magical moment because there were so much database talents gathered together into one place. What was that like?

Well, as you can imagine it was very exciting. It was wonderful at the beginning. Also, it was a different lifestyle in the sense that the environment was very sociable. You know, in universities people seldom share lunch with colleagues, or have parties with colleagues. There, it was different, perhaps because we were all new. And I remember having some wonderful experiences with friends and colleagues, including tennis games and windsurfing.

So tell our readers who was there.

Well, for instance, Patrick Valduriez and François Bancilhon, were there from France and among the people I used to socialize with; but there many others, including Mimmo Saccá, Fosca Giannotti, and Sergio Greco from Italy. Dick Tsur came from Israel and so did Oded Shmueli, and Haran Boral, a former PhD student of David DeWitt. Some remarkable young people also came through there, including Raghu Ramakrishnan (then a student at UT), Gerhard Weikum, and Mike Franklin. Mike was at MCC at the beginning of his career, before he went for his PhD. I don't know if Mike will agree on what I am saying, but it was probably that lifestyle and excitement which

[5] Carlo Zaniolo: The Database Language GEM. SIGMOD Conference 1983: 207-218.

enticed him to go for his PhD – and that was obviously the right choice for him, right?

It's worked out well for him.

Pretty well, in fact, extremely well, right?

[...] identifying the best students that can contribute to your research can be a challenge in a university environment.

You mentioned that the transition to academia was difficult. What made it different? What is it about being in the academic environment, compared to MCC, that is a challenge?

Well as you know, at universities, it is hard to get resources. One has to struggle to get funding and, of course, that was something I was prepared for. But identifying the best students that can contribute to your research can also be a challenge in a university environment. I mean, in a top research institute one can select people with a proven record of accomplishments and hire them by offering a good research environment and money. Also one knows the specific talents of people, and in particular who is a good system person. But initially a professor does know for sure the best talents of particular students when he/she starts working with them. So it takes time. For me, also preparing courses took time, everything took much effort. But, as I was telling you, things get better with time, not worse. I'm sure this resonates with many of my colleagues.

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

Well, I can tell you what worked well for me. When looking for a PhD research topic, a young researcher should focus on new areas. Thus one should try to find a new area of opportunity, rather than pushing the frontier in established areas which can be difficult because the pace of progress in Computer Science is very fast – in many ways it is even excessive. Indeed, over lunch with my colleagues, I often joke and say "We are doing everything wrong, you know? With a slower rate of progress we could have placed three or four generations of successful researchers, making slow progress with contributions broadly recognized. Instead, we burnt through our best opportunities during the last thirty years!" Naturally, this fast rate of progress makes some researchers feel kind of obsolete, right? I mean, look at other modern technology fields like aviation engineering. Eighty or seventy years of progress is not a long time for this and other advanced fields to see their technology mature. Instead, in the computer field we are trying do everything in twenty or thirty years.

So, because the fast pace of our field, young people should probably try to go into a new area and select new problems to work on. At the same time, however, our field is too easily distracted by new areas and we leave behind some of the tough problems that have emerged and remained unsolved in the course of previous research. I mean, that taking the low-hanging apples is very good for young people, because otherwise they will not be recognized. At the same time, established researchers who stumble on important hard problems should not give up easily. They should continue working until they get some real progress, perhaps some breakthrough result. That is why I was telling you of my interest in logic, where we have some non-monotonic reasoning problems which have remained unsolved for almost forty years, and where I hope that the Datalog research will soon see some breakthrough result that will stay with us for a long time.

In summary, what I have been trying to say in my long discussion is the following: if you're young, go for the new things. But, if later on, you find key technical problems that are very interesting, as a researcher you should have the pride to keep working on them. Furthermore, one should never turn down good papers simply because the topic is no longer in fashion.

So when you spoke of areas that we've left behind too quickly were you thinking of logic there?

I see logic taking on an important new role: it has been turning into a programming paradigm and a very exciting one. We have a tremendous amount of data, and we need general-purpose ways to handle it. We might want to handle small data on personal computers, and handle massive databases on large parallel systems, and we might also have to support data streams. So, we must make the techniques and algorithms we use highly portable, and for that an extreme level of declarativeness in the application language is needed. I think that, up to date, the logic of Datalog is still the best way to achieve a very declarative application language for big data. That's my viewpoint.

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

Yes, as it should be obvious by now, I am partial to my work on non-monotonic reasoning, particularly that on the choice models and XY-stratification. This work is being rediscovered now, and then there is the new work on aggregates in recursion .

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

You're talking about some technically related things?

For some people the answer is technical and for some it's non-technical.

Well, on the technical front, I'm very intrigued by what is happening with Wikipedia and how it is changing the way people gain new access to knowledge. This is not the first such revolution: the first encyclopedia in the 18th century changed the world in many ways. Likewise, people did not expect Wikipedia to change the way we do research, but it did, and it is also changing the ways we share knowledge. I got very interested in finding better ways for users to query Wikipedia and we proposed something very simple. We activate the Infoboxes in Wikipedia pages, to allow users to enter query conditions that are translated into SPARQL and executed on the DBpedia KB. Thus the user gets back all the pages which are relevant to that query in a very precise way. For instance we can use "not", or "greater than" conditions, which are very specific conditions that free-text search engines do not support well, at least for now.

That was on the technical side. On the non-technical side I have two granddaughters, and I would like to spend more time with them.

Are they in Italy or in the US?

They are here in the US, they live within driving distance from us.

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

I should be less selfish. So far, I have been involved with my own discipline, and that has taken most of my time and attention. If I had a chance to do it again, I would like to spend more time with scientists from different disciplines. That would not only enrich my view of the world, but also give me opportunities to be more effective as a computer scientist. It's obvious that advanced knowledge-based applications are now driving our field, and that is where we will be going in the foreseeable future. So, I encourage my colleagues not to do like me and just work on the most interesting problems: computer scientists should also think more on how new solutions can be applied to real world problems, and communicate more with real people and scientists from other disciplines to see what we can do to help them in solving their problems.

Thanks very much for talking with me today!

It was a pleasure.