# *Michael Franklin Speaks Out on Data Science*

**Marianne Winslett and Vanessa Braganholo**

## Mike Franklin

https://cs.uchicago.edu/people/profile/michael-franklin/

*Welcome to ACM SIGMOD Record series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we're at the 2017 SIGMOD and PODS conference in Chicago. I have here with me Mike Franklin, who is the chair of the Computer Science department at the University of Chicago. Before that, for many years, Mike was a professor at Berkeley where he also served as a chair of the Computer Science division. Mike was a co-founder and director of the Algorithms, Machines, and People Lab, better known as the AMPLab. He is an ACM fellow, a two-time winner of the SIGMOD Ten Year Test of Time Award, and a founder of the successful startup, Truviso. Mike's Ph.D. is from the University of Wisconsin Madison. So, Mike, welcome!*

*Everyone wants to know why you moved from Silicon Valley, the epicenter of all things computer, to the Midwest?*

I had a great 17 years at UC Berkeley and being in and around Silicon Valley. But I moved to Chicago to take advantage of an amazing opportunity here to help build computer science and data science in a new way that's integrated into the fabric of the university. The University of Chicago has tremendous programs across a huge range of fields, ranging from biological sciences to public policy, economics, of course, social sciences, and humanities and they've decided as a university that they want computer science and data science to play an increasingly central role across all those different fields.

And so, the opportunity I have at Chicago is to build a modern Computer Science department that in its very nature is built to work with people across all these different disciplines. And that combined with the opportunities of the growing tech field in Chicago and the Midwest more generally, it just seemed that after a great run at Berkeley it was time to do something new and so that's what I signed up for.

*What do you miss most about Silicon Valley?*

Silicon Valley is really a unique place. The energy, the sense of adventure, the sense of just trying to make something big happen that permeates the whole place is something that is hard to replicate somewhere else. So, I miss that, yeah.

*You miss it, but it is a bubble.*

Yeah, so the downside of all that energy is it's really all-encompassing. And when you're there, you're out talking to people, you're sitting in a café, you're at a restaurant, the topic of conversation is stock options and the next round of funding and the minimal viable product and all this. At some point, it does get to be a bit much.

One of the great things about Chicago and the Midwest in general is it's a much more diversified place. There's no one industry that dominates the Chicago economy and because of that you get people with widely varying interests all talking together, working together. It's, in some ways a refreshing change.

*You weren't very bullish on the short-term prospects for real-time streaming analytics when you gave a keynote on that topic at a VLDB 2015 workshop. Have your views shifted in the two years since then?*

So, I think some people might have misunderstood what I was saying in that talk, and I've given versions of that talk in different places. So, just for some history, we were working on streaming in the early 2000s when that was a hot topic in the database community, and the company you mentioned, Truviso, was a streaming analytics company. My view has always been that stream processing is absolutely going to be an integral part of any data analytics platform because it's just the most efficient way to answer queries that you already know you want to ask.

If you already have a bunch of queries, which often you do, that you know you're going to want the answer to, it's much more efficient to answer them incrementally on the fly as the data's arriving as opposed to storing the data off and then going to find it later and then starting the query from scratch. So, my view has always been that streaming would be a component of analytic systems. Now, what I did push back on is this idea that everybody is going to want instantaneous answers to queries, no matter what they're doing, and that's just not the way it turned out the first time we did it, and it's not going to turn out this way either, and there are a few reasons for that.

One reason is: business processes and other types of processes just have a natural cadence and that for a lot of reasons you can only make decisions every so often. And getting people the freshest data instantaneously when they're not able to make a decision or act on a decision is, in the best case, a waste of time, and in the worst case, a big distraction.

The other problem is, if you're trying to answer queries instantly when the data comes in, you don't have time to deal with problems in that data. So, for example, if there's out of order data, which is very typical in streaming environments, if you're trying to answer things instantaneously, you're going to miss a lot of data. If there's an error in the data and you're not able to analyze it properly, you're going to cause problems that way. So, really, my view on it is that if you ask people, do they want faster query answers, they're going to say yes. If you ask people, "for the problem you're trying to solve, how often do you need a correct answer," you'd get probably a very different and in many cases a much slower rate. So my view on it is that stream processing is really important because it's a fundamental way of dealing with large volumes of data, but you've got to take into account what people are actually trying to do and then target the solution to the latency needs of that application.

*That's true of life in general, wouldn't you say? You need to have situational awareness. Too much situational awareness is bad; you end up with helicopter parents and things like that. And too little can lead to a disaster because you don't know what's going on. So,*

*it's a lesson that we need to think about in applications for data.*

Yeah, I think that's right.

> ***[…] getting people the freshest data instantaneously when they're not able to make a decision or act on a decision is, in the best case, a waste of time, and in the worst case, a big distraction.***

*The AMPLab is about algorithms, machines, and people. Unlike the other parts of the AMPLab, the people component hasn't produced results that have made their way into industry. Why is that?*

Right, that's a great question. So, the premise of the AMPLab when we started it was exactly that. If you're going to try to make sense of big data, you have three types of resources you can use. You can use algorithms in terms of machine learning and statistical processing. The machines part of the agenda was about cloud computing and cluster computing and basically throwing more scalable hardware at the problems. And the people part was initially about crowdsourcing and about bringing people to bear on the parts of the problems that weren't adequately addressed by the algorithms and the machines. The dream was that we would build an integrated system that combined all of these.

Now, what happened was certain parts of the AMPLab agenda just took off like rocket ships. The one that, of course, is most famous is Apache Spark and all the things around it. Apache Spark and its ecosystem has taken a leadership role, not just in academia, but more so in industry in terms of what people are doing with big data. And so when people look at the AMPLab, they see the Spark part of the agenda, and they say, wow, that was a huge success and really beyond what you'd expect from an academic project. But when you look at some of the other parts of what we were doing, they didn't have that same industrial impact as your question says – at least they haven't had it yet.

But one thing I like to point out (it's a little defensive about the people part of the agenda) is if you look at what happened for the people who worked on that part of the agenda, we had best paper awards, we had a series of papers including, I believe, the most referenced paper in SIGMOD from the previous five years[1]. The students and postdocs who worked on that project are now at some of the top universities in the country. So, by any metric, the people part of the agenda was a huge research success, but when you stand it up next to Apache Spark, it doesn't have that same industrial impact, which was your question.

So, now the question is: Why is that? There are two things. One is the nature of the way people think about people in an overall systems architecture. If you look at large web companies that are ingesting and trying to make sense of lots of data, and you ask them to draw out their systems architecture, you'll see racks of machines running certain processes and communicating in certain ways. All of those companies have armies of people that are doing exactly what we set out to do in the AMPLab, which is to have the people do those things that you just couldn't get the right fidelity out of the algorithms and the machines to handle properly. But nobody draws their architecture saying, "oh, and the hard stuff that we can't afford to get wrong, we're going to show to this group of 500 people that we're paying". And so part of it is just there isn't the set of systems abstractions yet for how people fit into the architecture. Everyone thinks about it as those people are somehow separate from the architecture.

*Well, yeah, now that Facebook has to deal with the fake news in a very people-intensive manner, do you think that that will cause a change in people's thinking of what an architecture consists of?*

I'm not sure because they've been doing that all along. Companies have been bringing in people for solving those hard problems.

*True, but 2,000 of them, I think that's the number.*

Right, yeah.

[1] At the time of the interview (2017), this was referring to: Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin: CrowdDB: Answering Queries with Crowdsourcing, SIGMOD Conference 2011: 61-72. As of the time of publication (2019), a different AMPLab paper now holds this position: Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, Matei Zaharia: Spark SQL: Relational Data Processing in Spark. SIGMOD Conference 2015: 1383-1394.

*Do you think that will show up on future diagrams?*

I don't know what it would take to make it just standard practice to have the people part show up on diagrams. I think if there are enough stories like that where it starts coming out more in the open, maybe it will.

The other reason I think that industrial impact of the research has been a little slower than in other areas is because we don't yet have the relational algebra of crowdsourcing. We don't have a standard set of abstractions, of operators, of benchmarks that you need to make it so that people in industry can get their heads around a particular set of concepts and then start understanding what their needs are and what the alternative solutions are and how they compare. So, part of it is just this awareness about the fact that people are already integral parts of these systems. And the other part is that we in the research community have to do a better job of determining what are the fundamental abstractions of crowdsourcing to make it happen.

*What's next for you in your research life?*

Well, so I've taken on a pretty big administrative role these days, and I'm focused on basically doubling the size of our department. We're building a new building[2], we're building a lot of new programs, and so I think that's going to keep me pretty occupied for a little while. But as I talk to people around campus about what their needs are for data science, it's a lot of those fundamental problems that the SIGMOD community has been beating their heads against the wall on for a long time that still need to be addressed: data integration, data cleaning, and data quality.

There's an improved awareness now of how bias and other problems in the underlying data can impact the results that are coming out of analytics, and so really what I want to focus my research on, for the next wave of my research, is those types of data quality issues.

*You are currently building up a big new data science center in Chicago. How do you decide what to include in the scope?*

The challenge of data science is exactly that. That if you look across the disciplines in a modern university, pretty much all of them are doing more and more with data and feel that they need to be involved in data science. And so I've been taking a pragmatic approach, looking to see who's willing to step up and contribute and basically using that as my guide for what to keep in because intellectually, you really can't make an argument that these people are doing data, but that's not data science whereas these people are. But I think finding out who really wants to collaborate, who's willing to put some hard work into thinking about curricular issues and putting some time into it, that's what's going to drive who's involved, at least in the beginning.

*Could there be a problem later on if initial successes make a bunch more departments want to get on board, but you're built up or scoped out or whatever?*

I think whatever we design is going to have to be designed with the assumption that eventually everybody is going to want to be involved in some way.

> ***[…] there isn't the set of systems abstractions yet for how people fit into […] [a system's] architecture. Everyone thinks about it as those people are somehow separate from the architecture.***

*The University of Chicago has a great books curriculum. What do you think are the equivalent of great books in computer science?*

The University of Chicago curriculum, as far as I understand it, has moved away from the great books a little bit and is more of what they call now as a core. The core is determined less by specific titles and more by concepts and techniques and philosophies that you need to understand to be an educated person in the 21st century. And if you look at it from that point of view, I think it's pretty obvious that to be an educated person in the 21st century you need to understand something about computation, right? – this idea of computational thinking and how algorithms work and what they are and what they can do and what they can't do. And also you need to understand data, and you need to understand how to make arguments given data. You need to understand when somebody's presenting you an argument that supposedly comes from data, how to determine what might be right or wrong about that, and so that's more sort of a data literacy. If you think about

[2] The U Chicago CS Department moved into its new facility in August 2018.

computational thinking and data literacy, I fully believe that they will end up in the core set of things that the university decides all undergraduates need to know, and I think that's going to happen everywhere.

*Do you think there'll ever be classics?*

I think that there certainly will be classics about understanding the relationship between computation and thought, right, and around the limits and the opportunities of artificial intelligence and things like that. If I had to give everybody a book to read, it would probably be the Gray and Reuter Transaction Processing book.

*I thought of that as a possibility, but you give it to the average or even a computer science major, and they're going to be "oh my God, I had no idea, I don't want to know."*

Right, so that's the problem. The amount of things you need to learn and do until you can understand what's in that book is too high.

*The level of complexity is kind of astonishing.*

Yup.

*Alright, we'd like to hear your advice for having a successful startup in the data space beyond all the advice we can already find on the internet.*

Okay, well, I'll say a couple of things, and you can tell me if it's already out there. It probably is.

*Okay.*

Related to that, one piece of advice I got when I started my company was that you're going to find people who have a similar or the same idea that you have and that's actually not a bad thing. If you're the only person who's come up with an idea in a hot area, there's probably something fundamentally wrong with it. And so, if I'm giving you advice, it's probably already out on the internet somewhere or it's wrong advice.

But the thing that I learned about data-driven startups that surprised me was – you know, our first customer in our company was a hedge fund that was doing currency trading and we built an amazing application for them that let them see things in real-time that they couldn't see before. And in that business, you really do need to see things in real-time. We had that running, and we then went over to have a meeting with a computer security company, and they said, "oh, well, do you have a demo of your product" and we said "absolutely". Then we showed them the demo of this currency trading application. And they said, "well, what's that?", and we explained it. They said, "well, what does that have to do with computer security?" We said, "well, you understand, right, there's streams of data coming in, there are these comparisons and metrics and thresholds and other things are being computed in real-time and they're being shown. So, you can imagine that these are now network events coming in and the things that you're querying are security events…" But they just totally couldn't get it – and these were smart people, these were not dumb people. And then this repeated in a bunch of other industries as well.

*What if you'd framed it as situational awareness because security community does have that concept?*

It could be that we were missing the terminology, but really, I think what I learned from that is as data people, we're able to think about data and queries in the abstract – we see patterns, we see similarities. A customer that's trying to solve their problem sees only their problem, and they see it in the set of data that they deal with, and the set of questions that they ask. It's extremely rare, if not impossible, to find somebody at a company that's trying to solve a problem that has that same idea of abstraction that a database person would have.

So, my advice for people doing data-driven startups is to really put yourself in your potential customer's shoes and gear whatever you're presenting to solving their problems, not a bottom-up way about the technology itself.

*Interesting, so in the particular example you gave, does that mean you need to cobble together a fake network monitoring demo for them to get it?*

Yeah, I think cobbling a demo or…

*That's a lot of work!*

It's a lot of work, and I guess a corollary to that is you should very quickly pick a small number of verticals, maybe just one, and focus your energies on that for exactly that reason.

*Times have definitely changed, so do you think if you come back to a big network-oriented company today with your saying "here's your real-time dashboard situational awareness," would that now be an easy sell or would it be the exact same thing all over again?*

I think there's increased awareness. I mean, the story I told is probably a ten-year-old story. So, you're right, in the past ten years it's possible that there's a greater

appreciation now for data science in general, but I still think at the end of the day things that are very natural for you as a database technologist are not natural for a domain expert, and you just have to be aware of that.

> ***[…] we don't yet have the relational algebra of crowdsourcing.***

*You spent years working at MCC before you went for your Ph.D. Did you plan to get a Ph.D all along?*

I actually bounced between industry and academia quite a bit during my career. I worked after my bachelor's degree and then the MCC job was after I got a master's degree. I really had no intention of getting a Ph.D at the time when I took that job, but I worked with a lot of Ph.Ds and during the course of that project, we built a system called Bubba[3], which was one of the first massively parallel database systems. And working with the people who were Ph.Ds kind of impressed me. I liked the way they thought, I liked the way they thought about problems, and the way they attacked big problems, and it kind of gave me an appreciation for that and realized that that was something I wanted to learn how to do.

*Now, that's interesting. So, I guess master's degrees were the highest degree offered by where you got your master's degree. Do you think if you'd been at a place that also had Ph.D students you would have had that same reaction at that time?*

That's a good question. I always encourage people at universities now (especially undergrads), to do some research, to get involved in a research project to see if they like it, because you're right, had I been at a place that was actively doing research projects, I might have had that same experience.

*Interesting. Did your time away from the university affect what you ultimately chose to do for your Ph.D?*

Oh, absolutely. So, at MCC, in addition to working with some amazing people at MCC, I also had the opportunity to work with Mike Carey who eventually became my Ph.D advisor and Dave DeWitt, both of whom were consultants on that project and they were the reason I went to the University of Wisconsin.

*Most people go straight through school for a variety of reasons. For people who think they want to get an additional degree, under what situations do you think they should spend time away from school first?*

Well, I often encourage people – if you're in a place that has a one-year master's program, there are a number of schools that you do your bachelor's, if you stay another year, you get the master's… I think that's a really great opportunity for people and I also encourage people to do that. To go for your Ph.D, that's a whole other thing. One thing that students who are considering a Ph.D don't often understand is that everything up to the Ph.D, often including the master's, is basically course driven, so you take your courses, you do the exams, you do the projects. At the end of the semester, you're done, and you move on.

As you know, a Ph.D isn't like that at all. It's kind of an unbounded enterprise that you're getting involved in and it's a very different set of criteria for success. And you often don't get that regular feedback, that regular sense of accomplishment – certainly in the first few years when you're just trying to find your way around the research world. So I always encourage people, even who think they want to get a Ph.D, to maybe take a little time and spend some time in industry and see what's out there, to see if that's going to be a better path for them. My feeling, and it's just based on my own personal experience, is as long as you're not out too long, if you really want to go back, you will, and there will be opportunities to do that.

*What approach to advising do you take with your own Ph.D students?*

My approach with students has, I think, tended a little more towards the hands-off approach. I try to give students a direction and pose problems and then turn them loose on those problems and not dictate what I think the solution should be.

When I first became a professor, I ran into Jeff Naughton, who was one of my professors at Wisconsin, and he had asked me how it was, was I enjoying faculty life and whatever. I said, well, I really am, except that I really wish I could figure out what it takes for a given student to be successful because things that work for one student don't work for another. There are some people who need a little bit of pressure, there are some people

[3] Haran Boral, William Alexander, Larry Clay, George P. Copeland, Scott Danforth, Michael J. Franklin, Brian E. Hart, Marc G. Smith, Patrick Valduriez: Prototyping Bubba, A Highly Parallel Database System. IEEE Trans. Knowl. Data Eng. 2(1): 4-24 (1990).

who crack under pressure… I haven't figured out an approach to make students successful in general. And Jeff, who had been teaching for a number of years at that point, said to me, well, yeah, when you figure it out, let me know.

So, I think you do have to understand that people are motivated in different ways, but the best way to work with me is to be open-minded about the problems that you're going to work on and then be creative about the solutions you come up with.

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

We're really lucky in the database world. Those of us who have been in the business for a long time remember when it was one of the less glamorous parts of the field, where it was hard to get people to take the classes, it was hard to get people to want to do research compared to some of the sexier parts of computer science. But because of the big data revolution and because of data science and all the companies that are clearly being driven by data, that's not true anymore. I guess my advice is really to just enjoy being in a field that's having such an impact on the world and that has so many open problems.

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

Well, the AMPLab is hard to beat as a research project for a number of reasons. One, the impact that we had was really just, as I said, well beyond what you would ever hope for from an academic project and that's been really wonderful. But the other great thing about the AMPLab was that it was a collaboration of a large number of faculty, a large number of students, and these were people not just from databases – actually not even just from systems, but we had machine learning people, we had HCI people, we had security people, and then we had applications people around the campus who we were working with. So overall, it's hard to beat that experience as a research project.

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

I haven't written a book yet and I think I need to write a book, so if I magically had time, that's what I would probably do.

***I fully believe that [computational thinking and data literacy] will end up in the core set of things that the university decides all undergraduates need to know.***

*What would the topic be?*

Well, clearly, we need a new database textbook for undergraduates and somebody has to do that. The other one would be about building systems for large-scale machine learning.

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

If I could go back to my education, there's definitely some math classes I would have paid more attention to, because I'm finding now that a lot of the techniques that are taught in those courses are more important to me as a database researcher than I thought they would have been.

*Is that, in some sense, a failure of the professor or was the class so generic that the teachers themselves could not have envisioned all the different ways that their students might put that work to use?*

Well, I think, particularly for databases, advanced analytics and machine learning have now become so important that a lot of techniques around linear algebra and stochastic processes and optimization are just much more important to the field than they were even ten years ago.

*Okay, great, thank you very much for talking with us today.*

Thank you.