Juliana Freire Speaks Out on Reproducibility and Hard Changes

Marianne Winslett and Vanessa Braganholo

Juliana Freire

https://vgc.engineering.nyu.edu/~juliana/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I am Marianne Winslett, and today I have here with me Juliana Freire, who is a professor at New York University. Juliana is an ACM Fellow, and she has a Google Faculty Research Award, an IBM Faculty Award, and an NSF Career Award. She is also the chair of SIGMOD, and her term of office ends in just a few days. Juliana's Ph.D. is from Stony Brook. So, Juliana, welcome!

Thank you so much, Marianne. Thank you so much for actually doing this great service to the SIGMOD community. I know for a fact that this series that you run is one of the most popular sections of SIGMOD RECORD. So, thank you so much for doing this.

It's a pleasure.

Your colleagues say that you have been quietly battling against outdated traditions in the database research community for a long time. What have been your biggest battles and biggest accomplishments there?

One thing that I've learned in all these years that I am in academia is that change is difficult, and change takes time. At SIGMOD, we wanted to make some big and some small changes. And some small changes actually turned out to be big. For example, to have a diversity of opinions, gender, geography, as well as cover more areas, we proposed changing the structure of the conference to have two co-chairs. We faced a lot of resistance. Now after two rounds of SIGMOD with two co-chairs, the feedback from the chairs, program committees, and authors has been overwhelmingly positive. So, I think that this is one example of a small thing that turned out not to be so small.

Another challenge has been to increase the adoption of reproducibility in our community. This is something that my colleague Dennis Shasha started in 2008. And we have been making baby steps since then. There is still a lot of work to be done on this front. But I guess we can talk more about this later. Yes?

Yes! Since Computer Science moves so fast, why do we need reproducibility?

It is precisely because Computer Science moves so fast that we need reproducibility. If I do some work and you want to build on my work, how are you going to do that if you cannot reuse and extend what I did? If you have to start from scratch, this is actually going to slow down Computer Science. Reproducibility is necessary, specifically to make it possible for science, and Computer Science to move forward.

Do you see circumstances where reproducibility might impact science in a negative way?

I can't think of how reproducibility can be bad for science. There are some barriers to reproducibility. For example, works and experiments that use private data or proprietary software can be difficult or impossible to reproduce. People also cite, for example, intellectual property as another barrier. More recently, there have been concerns about open science and reproducibility being misused by bad actors. So maybe that would be one potential negative side of reproducibility, but for society in general.

Should privacy, the right to be forgotten, factor into how reproducible science is maintained?

There can be privacy issues in the data that is used in a particular scientific result and that must be respected. But there are ways of mitigating this problem. There are people working on synthesizing datasets that have similar properties but that do not disclose personal information. If you are talking about privacy with respect to "Oh, I did my work, I published it at SIGMOD, I have my experiments, but I don't want anybody to see those experiments." then, I disagree because I think that science has to be open. If I have my results, in particular, if my research was funded by the federal government, it was paid by the taxpayers, I have no reason, no good excuse, not to actually make that available and open to everybody.

Another potential issue is that it's hard to keep a piece of software working in the long-term because the hardware underneath changes, the OS, and the libraries. Do we have a moral responsibility to keep our research artifacts working, and if so, for how long?

Yes, I think that this is an important topic of discussion, in particular because there are costs associated with this. Lots of people keep asking how much should we actually invest in keeping old work as opposed to funding new research that is going to lead to new results. I think that the new developments around virtual machines and all the infrastructure that we have right now with the cloud make it a lot easier to preserve these research artifacts – to increase their longevity and make them usable in the longer term. This is definitely easier now. We should not aim to have these artifacts living forever. But I think it's important to try and keep them, for as long as possible.

There are efforts that aim to preserve such artifacts. Software Heritage is an initiative, started by Roberto di Cosmo at INRIA, in France, that is collecting all pieces of software that have ever been produced in the world -- you can think of this as software archeology. The goal is to have them forever, whether they are going to be running forever, that's a different question.

If I am running in a modern environment and I want to build on top of something that is living in a virtual machine from the past, how do I do that?

Depending on what you want to do and what you need to do, it can be easy or hard. Nowadays, there are workflow systems that allow you to stitch together different virtual machines. So, if the work is self-contained and you just need to input something and get some output, that's trivial. If it requires modification to the code and integration with new libraries, then, it can be very difficult. But if you have the software, and ideally the source code, it may be possible to more easily adapt it than to build everything from scratch.

> ***[…] science has to be open. […] if my research was funded by the federal government, it was paid by the taxpayers, I have no reason, no good excuse, not to actually make that available and open to everybody.***

Provenance tracking, being able to tell what information a particular conclusion is based upon, is super important for scientists. Does it matter for other people?

Of course! Provenance and reproducibility are now applicable to everything. We are witnessing a data and computing revolution: everything that people do now in government, industry, and science is around data and computing. More and more, decisions are being made based on results and insights that are obtained from data and computations. Provenance is key, particularly if you are making important decisions that have serious consequences. You need to be able to know what you have done, and reason about what you have done to make sure that you can build trust in the results on which you base your decisions.

I think a great example of that would be that the CDC said the chance of catching the coronavirus outdoors was 10% or something like that. Journalists traced that fact back in the data and found out it was based on data from construction workers in Singapore. Being someone who's lived in Singapore for a long time, I can promise you they didn't understand what construction was about in Singapore. So, the conclusion they made was erroneous. But on the other hand, despite the fact that they could have traced it back, they probably would still have reached the same erroneous conclusion, wouldn't they?

I think that there is deeper issue here related to meta-studies**.** People collect data for different purposes and meta-studies attempt to combine them to synthesize new knowledge and draw their conclusions. The problem is that the context and assumptions that are made for each of the different projects and underlying data used in a meta-study can be different, and inconsistent. It's difficult to reconcile all of those, and I think that is what happened in the study you refer to. Because it's a construction site, but it was not necessarily enclosed. I think that was the issue, right?

I think now, the real issue which most people don't know is that the construction workers in Singapore live together in dormitories with like 12 people to a room in bunk beds. It is the closest packed environment that you can imagine. So, of course, the coronavirus is going to spread under those conditions. But they didn't think about that. They imagined that it was always caught at work.

But then, this is an instance where proper provenance was not actually captured. Because if we had correctly captured the contextual information where the data was actually gathered, you wouldn't have had that problem. But in practice, this is difficult to avoid. You cannot avoid all of these mistakes or oversights. This is why it is essential to have transparency and be able to trace back the steps. In this case, the journalists were able to go back and look at the data and figure the problem out. You need to capture as much provenance as you can to enable you and others to go back to assess and debug the results.

Your open-source workflow and provenance tracking system, VisTrails, was ahead of its time in many ways. What about its impact are you most proud of, and what lessons did you learn from that?

I think that VisTrails was my first project that had real practical impact. It ended up being widely used by many different people, different communities. Big projects adopted it. And there are lots of things that contributed to that. First, we had a great team working on the system. We had a group of Ph.D. students that were not only talented researchers, but that were also amazing hackers and very passionate about the project. VisTrails was written and rewritten about three or four times. And if you look at the code, it is professional. The system worked, and it worked well. An important lesson that I learned is that if you want to do something well, you

need to have the right team. And in this case, we were very lucky to have the dream team.

VisTrails is a good example of a multidisciplinary project. And for such projects to succeed, we also need to have the right collaborators. We were very fortunate to identify a number of people, including physicists, biologists, medical doctors, that worked closely with us and from whom we actually learned what the real problems were, what their real pains were. We designed a system to meet the scientists' needs. At the same time, because we were working so closely, not only did we solve their needs, but we also were able to get into a virtuous cycle: we solved the real problems that the scientists had, and at the same time, we found a number of interesting Computer Science problems. And this is how three different Ph.D. dissertations, and many papers, came out of the VisTrails system.

Another big challenge that we had was maintaining an open-source system at a university. Raising funds to support programmers (after the Ph.D. students were done) to actually keep the project going and supporting users is extremely challenging. We lack (both in funding agencies and at the universities) the proper infrastructure to keep research software engineers. This is a fight that I am still fighting within NYU. If we want to have successful Data Science, Computer Science applied to science projects, we need to have research engineers and proper career paths for them at the university – they are critical to the success of our research and need to be recognized as such.

At some point, you moved your focus from captured workflow to providing provenance support for Python scripts and Jupyter notebooks. Why is that?

This is another lesson that we learned from VisTrails. The project was very successful, but to use VisTrails and to reap up all the benefits that come from provenance that the system automatically collects, people have to adopt that system. There is not only a learning curve but also a ramp-up period in which you actually need to adapt your research environment and integrate it with VisTrails. For some people, that worked, but many people want to keep working with the tools that they are already familiar with. So, my vision was: "Can I get the same benefits of VisTrails, but within the environment of Jupyter, of Python, that tens of thousands of people actually use on a day-to-day basis?" Let's get reproducibility to the masses without having to put any burden on them.

That system, ReproZip, has shown itself to be really useful in creating reproducible artifacts. How did you come to develop ReproZip?

One of the key issues that we observed is that systems like VisTrails and other workflow systems, capture provenance for the steps that are followed in the workflows, for example, processing the data and building a machine learning model. If you have the specification, you can rerun those steps within the workflow system. The problem is that if I want to share them with you and you want to run those on your machine, you may not be able to because there are the dependencies, there are libraries, there are different Python versions, different scikit-learn[1] versions. ReproZip captures the provenance of your computational environment: everything that your experiment needs, files that it reads and writes, and libraries that it uses. It automatically creates a package that contains not only your computational steps but the whole environment required to run those steps. And once you have that, you can reproduce the experiment on a different machine or in different operating systems. ReproZip solves the dependency hell problem.

[…] we need to have research engineers and proper career paths for them at the university – they are critical to the success of our research and need to be recognized as such.

What if the artifact depends on old versions? Can you reproduce that?

Oh yes! ReproZip works as follows: when you run your experiment, it watches at the operating system's level everything that is touched and invoked by the experiment. If the experiment uses a specific Python library, ReproZip will identify the library. And when you create the package, ReproZip copies that library, the old library, into the package. Then, when the package is run within a virtual machine, you will be running the experiment exactly like it was run on the author's machine.

That can save a lot of pain.

[1] scikit-learn is a popular Python machine learning module.

Exactly.

What do you see as the future of tool-based reproducibility?

That's a good question. So, I can tell you what my dream is. My dream is that reproducibility will become a standard component of all computational environments. You should be able to work, do everything as you currently do, and with the click of a button, you will be able to retrieve everything that you did with essentially, zero additional work. This is what we should aim for. There has been substantial progress in the past few years, and nowadays, attaining reproducibility is much easier. There are lots of open-source tools, virtualization technology, clouds. But there are also gaps which can make the creation of reproducible results difficult in some scenarios. We need to better understand these gaps, and address a number of research and engineering challenges. I have been working towards convincing funders to have Programs to fill these gaps so that we can have reproducibility everywhere.

So, we might expect to see new calls for proposals that target those gaps?

If I am successful, yes.

> ***Let's get reproducibility to the masses without having to put any burden on them.***

You like to work on data management issues for emerging applications. What's the next big thing for the data research community in terms of applications?

There is a broad area of trust in the data and computation that I think is extremely important and has great potential for practical impact. And this ties back to what I mentioned that data and computation now are at the center of everything -- this sounds like a cliché, but it's actually true. As we have more and more people using computing and data, we need to have better mechanisms to guide them and help them build trust in what they do. We need to have better support for identifying issues, bugs in data, the computational steps executed, and in the computational environment – all of these can actually impact your results. This is a huge area with lots of very interesting research problems, and there is a huge unmet need for this right now.

Great, sounds very interesting.

There has been growing interest in machine learning models. You have your machine learning pipelines, and you want to explain the results for those pipelines. I think that we should be asking a broader question, in addition to machine learning, we should seek to understand and explain computations in general – Machine learning is just one component of the data science pipeline. How you obtain the data, what you do with the data, the kind of preprocessing, computations all contribute to the results produced by machine learning tools.

You have been the chair of ACM SIGMOD for almost four years now. What changes have taken place during that time?

I've actually been looking at some of the plans from four years ago, what I had in mind when I became chair – a retrospective look at what I wanted to do and what I actually did. One of the challenges that I identified is the fact that our community is growing and it's becoming more and more diverse. When I say diverse, I mean in all different aspects – not just demographics, but also in research areas. The status quo is that papers have to be about specific, traditional topics, for example, database engine. There is also a mindset for what a SIGMOD paper looks like. One goal that I had was to open this up. We are a big community – how can we actually let all flowers bloom? And how can we recognize all the different types of work? Our goal as researchers is to have impact and to have impact, we need to work on many different problems.

There have been a number of changes at SIGMOD that go in that direction. We have a new Applications track that aims to bring people from different areas to work with us, with our community, that was introduced by Divesh and Stratos and is now being refined by Amr and Angela. There has also been a lot of work by the PC chairs of SIGMOD to educate the reviewers to recognize different types of work and also review papers with a positive mindset, what AnHai and Wang Chiew termed as "review to accept". This is a step towards changing the culture that "we want these kinds of papers, and if a submission deviates, it is not worthy of SIGMOD." This requires educating reviewers to try and recognize novelty in different types of work, and contributions that will not only move our community forward but also lead to impact.

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

Choose the right problem to work on. Selecting a problem that matters and has potential for practical impact is very important (at least to me). And not only

that, choose something that you are passionate about because things are hard, and it is a lot easier when you are passionate about something to actually keep on it even when you fail over and over again.

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

It depends, Among my past projects, the body of work that we did on provenance and VisTrails is probably my favorite because it addressed an end-to-end problem, it involved theoretical and practical research, interdisciplinary collaborations. We went from the conception of the initial idea to doing Computer Science research, applying this research to different scientific domains, developing and deploying software. The work that I am doing now on building trust, debugging and explaining computations is something that I am very passionate about. It is at a very early stage, but it is a good candidate to become a favorite.

If you magically had enough extra time at work to do one more thing, what would it be?

I would spend more time working towards mentoring young minority students. I am Latina, and there are very few of us there are in academia or in top positions in Computer Science and Data Science. So, I wish I had more time to devote to increase the representation of minorities in Computer Science. I am making some time for this in the summer. NYU Tandon has a program called ARISE[2] that recruits high school students from underprivileged communities, and they spend a month at NYU. My lab will host two ARISE students. I hope to devote more time to this and similar initiatives in the future.

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

This is a tough question. Career-wise, I think that if I look back, I would probably have tried to plan more and be more strategic – things happened, and I just did it. Maybe my life would have been easier had I planned, but maybe it would have turned out differently, and I am happy as is.

Thank you very much for talking to me today.

Thank you so much, Marianne. Nice talking to you.

You're welcome.

[2] https://wp.nyu.edu/k12/arise