Beyond Academia

Heather Miller

The driving force behind everything that I do is a desire to affect the greatest positive change for as many people as I can. This desire informs not only my research and teaching efforts, but all projects I devote myself to. As a result, during my Ph.D., I spent significant portion of my time devoted to numerous endeavors which meaningfully impacted large communities of developers, students, researchers, and more, but were unrelated to my thesis. Ranging from organizing conferences for academia and industry alike, to sustaining the development of the Scala programming language, including Scala's image, documentation, and community. I sketch just some of these efforts below.

In short, I try to leave things in a much better state than I found them in.

Curry On Conf: Academia and Industry Should Talk More

In working on Scala, I've been fortunate to straddle the academic-industrial divide. Though, something that's long vexed me from this vantage point–industrial programming language communities often approach academia with disdain, and vice versa. Either that, or they tend to turn a blind eye to one another. One idea: perhaps a new prestigious professional venue aimed at bridging this gap could chip away some disdain and expand horizons.

To this aim, as a PhD student I created Curry On [1] with Jan Vitek, a new and unusual conference focused on programming languages and emerging challenges in industry. In particular, Curry On seeks to act as a conduit for ferrying understanding and ideas back and forth between industry and academic programming languages, software engineering, and systems research communities (amongst others). As the pitch goes, "Curry On is a unique event where academic minds responsible for concepts and tools now invaluable to everyday software development–like functional programming, or generics in Java–collide with the movers and shakers in industry that are building next-generation systems and developing software engineering practices central to our entire industry." Curry On is held in a different European city year-to-year, and is always co-located with one of the top academic conferences in programming languages–so far ECOOP and PLDI.

As Curry On attracts one of the most varied technical audiences to any one conference, questions are heavily encouraged–they are key to cross-talk between communities. Further, a special kind of talk, called a "chess-timer talk"[1] has been introduced in an effort to make the conference a more interactive, more fun, and better place for learning and discussions across the academia-industry divide.

To our surprise, Curry On quickly garnered widespread interest in both academic and industrial circles, going viral on social media and even selling out of tickets (unheard of in academic circles!) On average, around 50% of our attendees hail from industry, and around 50% hail from academia.

We're proud and fortunate to have hosted exciting speakers from some of the most successful bridges between academia and industry, particularly in the programming languages community. Some of our more recognizable speakers include:

[1] Speakers who choose to give a chess-timer talk are allowed 20 minutes of solo speaking time, and 20 minutes of discussion time. A Curry On representative operates a chess-timer during the presentation, switching between solo + discussion time budgets. When an audience member interrupts the talk to ask a question, for example, we switch the timer to deduct from discussion time.

Brendan Eich, *creator of JavaScript*
Simon Peyton Jones, *co-creator of Haskell*
Larry Wall, *creator of Perl*
Martin Odersky, *creator of Scala*
José Valim, *creator of Elixir*
Roberto Ierusalimschy, *creator of Lua*
Dave Herman, *Rust*
David Nolen, *ClojureScript*
Bjarne Stroustrup, *creator of C++*
Gilad Bracha, *creator of Java generics*
Philip Wadler, *co-creator of Haskell*
Matthias Felleisen, *co-creator of Racket*
Brian Goetz, *Java*
Stefan Karpinski & Jeff Bezanson, *creators Julia*
Evan Czaplicki, *creator of Elm*

Curry On has also gone on to double the attendance of ECOOP, and attracts around 300 attendees from across industry and academia each year. Curry On has taken place in Prague in 2015 [2], Rome in 2016 [3], and Barcelona in 2017 [4], with a fourth edition in Amsterdam in 2018 in the works!

The Scala Center: Sustaining Scala

On the one hand, it can be an unthinkable blessing for one's research get the industrial attention that Scala has received while a PhD student. On the other hand, popularity in open source can also be something of a curse.

It is no secret that in the early years of Scala's development, all engineering on the Scala compiler, standard library, and tools was done solely by Martin Odersky and his PhD students. It's also no secret that working on things like build infrastructure for a production compiler, compiler bug fixes, and important utility libraries for a programming language with hundreds of thousands of industrial users is not necessarily the most direct route to a PhD.

So, in 2011 venture capital (VC) was brought in to help. However, after a few years of a VC-backed start up taking over some of the maintenance burden of Scala, we realized that this too was not sustainable. VCs expect an explosive return on their investment, which isn't obviously obtainable from an open source compiler. At the same time, it's not uncommon for VC-backed start ups to get distracted from their initial goals, and to wander elsewhere looking for explosive returns, while still holding on to anything of value with a white-knuckle grip. This too wasn't an ideal scenario for this language which industry has adopted en masse and which served as such an excellent vessel for research. Progress had to be made on the Scala compiler and libraries, and ours hands were tied.

In early 2015, EPFL Direction came to us with a request; the Scala MOOC had made EPFL the leading university in Europe for MOOCs, and EPFL wanted to go to the next step by offering a Scala mini-degree on Coursera. Together, Martin Odersky and I came to an agreement with EPFL Direction–we would form an organization to concurrently support Scala research and development as well as the MOOCs. And thus, the Scala Center at EPFL [9] was born.

Alongside of income from the MOOCs, The Scala Center is funded by industry donations. Companies who have built the core of their business on top of Scala join as corporate members, and contribute a fixed sum *yearly*; currently ~$50,000 per year from 10 separate companies across vastly different industries. Interestingly, a requirement of membership is participation; each quarter, all members join a meeting where representatives from each company share the good and bad about Scala across their organization. These insights from large organizations have been invaluable!

Our mission? Invest 100% of those donations in bleeding-edge research engineering and sustainability efforts; e.g., tooling, compiler and library development, and community management.

On the technical front, we work on production-ready solutions to hard problems at the boundary of software engineering and programming languages. One example is automatic code migration–how we help users automatically upgrade massive codebases to new Scala versions? Or linting tools–Scala is an unopinionated language, but our users can be very opinionated. Can we develop robust and customizable linting tools that can be reliably run on Twitter's multi-million line Scala codebase? Or compile speeds–can we rethink incremental compilation to get better compiler performance? These are just a handful of the sort of hybrid research-engineering problems we focus on.

On the community front–the Scala Center was founded at a time when community trust in "the powers that be" behind Scala was at an all-time low, abuse was rampant in our open source community forums, and our contributor count was in a free-fall. We needed to earn the community's trust, and we needed to take better care of our community. To that end, we have gone to great lengths to democratize (1) language evolution, and (2) core library evolution through the introduction of new, transparent community processes. To deal with abuse, we helped put into place a community-developed Code of Conduct, and revamped our communication channels. To encourage more companies to give back by investing developer time in open source Scala, we devised and regularly organize "open source sprees"–events aimed at bringing project maintainers and potential contributors together, and at teaching how to contribute to open source.

Simply put, our mission is to be the vendor-neutral

steward of Scala, always careful to do The Right Thing™ for community and for the growth of the Scala project.

Did it work? We think so! In the 1.5 years since we were formally established, we've contributed heavily to existing Scala tooling, developed new tools such as linters and build tools, developed new core libraries, and went from our contributor count in free fall to growth.

Between MOOCs and industrial donations, the Scala Center supports around 5 engineers, a community manager, and an administrative assistant.

What's my role in all of this? Martin Odersky and I together conceived of and founded the The Scala Center. I defined The Scala Center's vision, mission, and strategy, established its unique funding model, and built the team we are lucky to have today.

Middling Workshop ⇒ Standalone Scala Symposium in 3 Years

From 2010-2012, a small and informal academic event called the "Scala Workshop" was tacked on to Scala Days–the main Scala conference attracting close to 1000 attendees from industry. Paper submissions and attendance were in free fall, despite co-location with flagship Scala Days.

It was an odd trend, especially given that Scala was rapidly being taken up in research groups around the world.

Taking the helm as principal organizer with Philipp Haller in 2013 [6] and 2014 [7], we reversed that trend. We separated the Scala Workshop from Scala Days, co-locating with ECOOP instead, and we published proceedings in the ACM digital library for the first time. Cognizant that workshops are of most value to junior students, we enacted a sponsorship program for students (B.Sc., M.Sc., beginning Ph.D.) with accepted short "student talks." It was a hit. Students earned the first technical speaking opportunity of their careers, and we raised over ten thousand dollars in sponsorship money that we used solely to fund around a dozen students to attend ECOOP.

The quality and number of paper submissions have since surged, graduating the event to the status of Scala Symposium in 2015.

Documenting Scala with >200 Online Acquaintances

When I joined the Scala group in 2011, Scala had just begun its rise in industrial popularity. Industrial users' list of Scala pain points preventing adoption was long. On the top of that list was documentation.

I devoted my first few months in our group to rectifying this. In addition to authoring numerous technical articles myself about Scala's most advanced features, I began a series of open source community initiatives to improve the state of Scala's documentation.

Most notable is Scala's "doc site" [5], an open source central repository for guides, tutorials, cheat sheets, and other documentation. I designed the initiative, curated documentation, and edited contributed material. The project was a success–to date, 235 individuals have contributed documentation, and the "doc site" receives around 200,000 unique monthly visitors while serving as the main entry point into learning Scala.

This community-based approach to growing documentation has even adopted by other popular languages, most notably, Clojure [8].

Core Contributor to the Scala Compiler, Libraries, Tools & Community Process

And of course, as a member of the Scala team since 2011, I have contributed to the Scala compiler, core libraries, core tools, and helped develop our open source processes. My contributions include service on two expert groups; (1) I am leading the effort to standardize my research on Spores, and (2) I have contributed to Futures & Promises, one of the most widely-used concurrency libraries for the JVM, powering production systems at Apple, Netflix, Foursquare, and more.

References

[1] http://curry-on.org.
[2] http://curry-on.org/2015/.
[3] http://curry-on.org/2016/.
[4] http://curry-on.org/2017/.
[5] http://docs.scala-lang.org/.
[6] http://lampwww.epfl.ch/~hmiller/scala2013/.
[7] http://lampwww.epfl.ch/~hmiller/scala2014/.
[8] https://clojuredocs.org/.
[9] https://scala.epfl.ch/.