

A wake-up call for the Logic Programming community

Tom Schrijvers

October 22, 2007

In the 6 six years that I have been doing research in the Logic Programming community, I have met a lot of nice people and heard about a lot of interesting research ideas. However, the community itself isn't exactly thriving and LP has a serious PR problem. It's my fear that if we happily continue along our current path, then there will be little left of LP and its flagship Prolog in a couple of years.

There are plenty of stories about the fabled heydays of Logic Programming with 300+ ICLP attendants, well before my time. However, nowadays, what are LP's claims to fame? How can we motivate people to join the field, while there are so many more exciting (better marketed) research fields? What career opportunities can we promise people who learn Prolog? What industrial applications can we mention? There must be some out there, but for some reason it is oh so difficult to list more than the one or two you always hear about.

The purpose of this article is to tell you about how the Functional Programming community, and in particular the Haskell community, has dealt quite successfully with very much the same issue.

My recent attendance of ICFP, the FP counterpart of ICLP, has been a real eye-opener in this respect. There are plenty of tell-tale signs of a successful PR machinery running at full throttle:

- There is an incredible boost in the number of ICFP participants: 240 participants this year, which is about twice the number at ICLP. There are about a 100 more than last year.
- The co-located Haskell workshop (one day, 10 papers) had about 120 participants. That's about the same number as the ICLP main conference! I'm sure it's 3x or 4x more popular than any ICLP single day workshop.
- The second most popular co-located workshop is the Commercial Users of Functional Programming (CUFP) workshop with 109 participants! Most of the speakers are indeed commercial users and not affiliated to a university.

- More than a few non-academics participate attend ICFP and its workshops. Commercial users and hobbyists feel that ICFP has something in store for them.

What's the secret of their success? These aspects must be having some effect:

- Functional Programmers are confronted head-on with the commercial applications of their work, and they cannot avoid being motivated. I already mentioned the CUFPP workshop. ICFP too puts a big effort in a special track of 6 experience reports. These are meant to build up a written body of evidence at the disposal of the champions of FP. Do not except any new research ideas or new insights in the use of FP. That's not what they are meant for. The purpose is really to reinforce the community!
- Functional Programmers know where to find a job when they finish their Ph.D. Companies are actively recruiting at ICFP. Get a free dinner when you sign up for an interview!
- A considerable part of the program consists of new ideas to make programming easier: libraries, tools and programming idioms. These are exactly what real programmers and companies need, in order to be convinced to adopt FP.

If the research community values talks about these topics too, then you get more research effort spent on them. More real world programming tools, means more real world programmers. More real world programmers means more researchers!

Of course, there is more to a community than just conferences and workshops, certainly if there is more to that community than just academics. Haskell is the undisputed flagship of the FP community. This language was designed by a committee and is especially welcoming to outsiders:

- There is a tremendous amount of activity on the Haskell mailing lists (Haskell and Haskell-Cafe): on October 10 I counted 83 messages versus the 7 messages on Prolog's main discussion forum `comp.lang.prolog`.
- The `haskell.org` portal website bundles almost everything there is to know about the language: papers, systems, tools, companies, tutorials and wikis with explanations about language features and idioms,...
- There is a central repository (hackage) for all kinds of libraries and projects.
- Missed a symposium or FP user group meetings to know what's going on? Simply catch up on Google Video! For example, all the Haskell 2007 workshop talks are on-line. In comparison, if found only two videos on Prolog which weren't even by LP people.

Everyone can step right in to contribute; in fact, the community insists that you join the fray. There is no central authority, no equivalent of the ALP, to

tell people what to do or what not. You don't need permission to update the website or to add another project to hackage. It works: the open source community around Haskell is growing and making Haskell into a viable alternative for mainstream languages. These days libraries are a major factor for adopting a language, and Haskell is getting them through its growing open source community, its Google Summer of Code projects and its hacking marathon co-located with ICFP, not to mention the researchers who can actually publish their work on libraries for real-world problems. The snowball was tiny when it was first thrown, but it's become a real avalanche now! And the Haskell community sure is fanning the flames.

Clearly this community is doing some things right by investing considerable efforts in community building. I think the LP community should start learning from this if it doesn't want to get extinct. There is little point in putting your hopes on parallel programming for making LP popular again, if the community does not start to value and advertise itself first. All the things I mentioned above, LP can do too:

- *real-world* application papers at ICLP and all these other signs that LP matters out there,
- talks on Google Video,
- more papers on programming tools and libraries at conferences,
- a centralized wiki website and project repository with all kinds of resources where we can find out about LP people, projects, software, companies, ...,
- a mailing list that pushes messages rather than a newsgroup from which you actively have to pull messages,
- the ALP and other sponsors should financially back community efforts.

All this should happen with the communities encouragement. Don't put up high barriers and high standards for people to contribute. Let them contribute first, and, if you don't like what's there, then improve it! It takes only a little effort from everyone to make a big difference.

There is one big hurdle that the LP community has. The Prolog implementations are hopelessly fragmented, and we cannot expect any hope from the ISO Prolog committee. We are in great need of an up-to-date language standard. It's been proven to be impossible to get everyone to agree. So let's not do that; let's lower our standards. Wouldn't it already be great if we could have JUST TWO systems that we could rely on to behave in the same way.

Here's my challenge for you Prolog systems out there:

- find a partner system,
- agree on a common module system,
- agree on a common set of built-ins, and

- agree on a common set of base libraries,
- present us your system at ICLP 2008.

Here's my challenge to the community to back the first two systems that manage to seriously cooperate:

- we conduct any generic Prolog research on these systems and say so in our papers,
- we contribute all the library code we happen to write,
- we teach our students to use these systems,
- we get our master students to write libraries and tools for these systems, and
- the ALP endorses these systems as the new Prolog standard.

Please let me and everybody else know what you think: tell us on comp.lang.prolog!