Conversation with a Prominent Propagator: Sushil Prasad

David P. Bunde
Zack Butler
Christopher L. Hovey
Cynthia Taylor

Developing new materials for, or approaches to, education is only part of improving CS instruction. In order to have an impact on students beyond those of the developer, educational innovations must be propagated to other instructors. Doing so has been shown to require significant time and resources on the part of developers [7]. We believe that the CS education community must embark on a sustained effort to learn more about evidence-based strategies for propagating educational innovations and to use them in our own projects. To support this discussion, we have recently written a review on research on propagation [8], much of it in other STEM disciplines. This column is our effort to turn the focus back to Computer Science and capture the knowledge within our own community through interviews with *prominent propagators*, those who have been successful spreading educational innovations to other instructors. In our first column, we interviewed Leo Porter about his work on Peer Instruction and spreading a variety of pedagogical techniques through the New Faculty Workshop [1].

In this column, we interview Sushil Prasad, Professor and Department Chair at the University of Texas at San Antonio. Sushil heads the Center for Parallel and Distributed Computing Curriculum Development and Educational Resources (CDER), which promotes the inclusion of parallel and distributed computing (PDC) in CS curricula. This group published a set of curricular recommendations [4], many of which were incorporated into the ACM/IEEE Computer Science Curricula 2013 [3]. They have also published 2 books of resources for teaching PDC [5, 6] and founded a series of workshops on PDC education that are collectively called the Edu* workshops: EduPar, EduHPC, Euro-EduPar, and EduHiPC, each co-located with a research conference on parallel computing.

Below are highlights of the interview, which ran approximately an hour. They have been edited for clarity and style.

Q: How did your project get started?

SP: I got elected in 2007 as chair of the IEEE Technical Committee on Parallel Processing (TCPP), and that led to my initial reach-out to the community. I sent out an email to the entire TCPP email list, and my question was, I teach this class every fall (autumn) on parallel and distributed computing (PDC) and the technology keeps changing, the books keep changing, and I didn't even know exactly what I should be teaching—what is current, what's not current? What should I do in architecture? algorithms? programming? and so on. So, how do we keep up? That email got the ball rolling, and the NSF funded a planning workshop in due course. That is how we came together.

The idea was that, now that the entire landscape has changed under us, everybody who's graduating with a computer science degree or computer engineering degree must know certain PDC skills. We took up this task of looking at the curriculum, to see what the “core” curriculum for PDC should be—”core” meaning the essential skills that everyone who is a computer scientist or engineer should have.

Q: When you first sent out that email, how many people responded?

SP: Oh, we got easily hundreds. It was wonderful to really see that response. And so, it was community-driven in that sense. It was a community need—people felt that it was needed.

We met in person at an NSF-sponsored planning workshop in February 2010, and we met again at the IPDPS conference in April 2010. And then we started meeting weekly from June to December, and we formulated, we debated. Most of the people involved were actually researchers in PDC—they teach, but they're not education scientists.

Q: How did these meetings lead to developing a curriculum?

SP: Everybody wanted to include their own pet things. But what is the minimum core? So, the debate was mostly about trying to *not* include everything, to find the essential core. We still continue with that mission.

Then we happened to get in touch with Richard Leblanc, who was working with the ACM and IEEE CS curricula committee. He informed us that if you want to do something real, you need to worry about specific things. For example, you cannot just say, “This topic should be taught.” At what level should it be taught? Where (in the curriculum) should it be taught? But especially, what is the required material?

So, we got the preliminary version of the curriculum released in December 2010. Then we sent it out for feedback. We got a lot of folks; it's not just a small committee. The entire community actually stepped up. We sent it out, we got a lot of reviews, and we refined it. All of that took time, I think a year at least. In 2012, we released a formal version 1. That is where the curriculum is right now.

Q: How did you identify who to send your preliminary version to? What was the selection process?

SP: I think it was mostly to people who were prominent in relevant content areas. There are four categories in this curriculum: architecture, algorithms, programming and cross-cutting topics, or things which are not necessarily just about programming or just about architecture. We selected folks in the community and then we approached them. It was not sent out to everybody in the beginning because we wanted to make sure that what we were thinking was reviewed by those who maybe know better or who have different institutional contexts. That was a first round and then we released it to the public and got some extra feedback and so on.

Q: How did you get your work into the ACM guidelines?

SP: We went to the SIGCSE community right away. There was a birds of a feather session at SIGCSE that was attended by members of the ACM/IEEE committee. They all actually had our curriculum in their hand and they were attending our BoF. The timing was critical because ACM was looking to identify the new areas that needed to be looked at in a big way. They came out with two main thrusts: one was security and the second was parallel processing. And so we could inform the ACM/IEEE curricula—at various stages they wanted our feedback. They knew what we were doing.

Q: How did you get people to adopt the new curriculum?

SP: At least three things, I would say. The first was to get a very diversified group of people in the working group as the key stakeholders. You don't want to just design something and say, “Ok, here it is.” People will pay attention if we have also paid attention in terms of connecting to all the key people. So industry, academia, authors, federal agencies, professional societies, these are the main players and we definitely needed to have them. The initial working group had that.

The second thing was awarding seed funding: $1,000 for a one course change, up to about $2,500 for an entire curriculum change. The idea was that instructors who are teaching, or a group of instructors who are teaching multiple courses, could propose what they're going to change and how they're going to do it. This was open to both international and domestic universities. NSF funds could be used only for domestic institutions, but Intel provided funds for international institutions. By the second half of 2015, we had at least a hundred institutions and 140 or so instructors participating. We have participation from all over: Singapore, Malaysia, New Zealand, China, India, the Middle East, Europe, and South America, in addition to the US. We were actually inviting them not just to use some of the material, but also to tell us how it went.

The third component was to start the Edu* workshops, where adopters would come and talk about things, and we also invited people who are not early adopters. That became a vehicle for the exchange of ideas, evaluation, and propagation—to have a community that started developing. The first was EduPar at IPDPS in Alaska 2011. It was so successful that the next year, IPDPS wanted it as a regular workshop, and that is special because research conferences typically don't have an education component.

In 2013, we started EduHPC at the SC conference and then we also started another sister workshop at EuroPar in Europe. Last year, we seeded EduHiPC in India. One thing that we have tried to do, especially for international conferences, is to try to get people from the region involved.

We had these competitions and workshops where we really reached out to the community and had them get involved. We reached out to all the projects that were already going on, like CSinParallel [2], and invited them to be part of this movement.

Q: What characteristics made adoption easier or harder?

SP: The connection to prominent research universities ("R1s") was important right in the beginning because we wanted to make sure that the original working group had prestigious names, with researchers who are well known. Involving them means increasing visibility, but then the idea was to reach out beyond R1s. The bulk of the early adopters are actually from non-R1 universities. So, broaden the participation and also involve non-R1s, especially the teaching schools because it's an enormous task to really change how things get done. I think it's all about the community, right? You need to create a community that understands and actually shares your vision and then wants to work together. I think creating a vibrant community for such effort is the key.

One thing that we have tried to do is to invite the community to contribute their syllabi and other resources for education at our CDER site. They also wrote book chapters. Apart from the workshops and early-adopter competitions, we are also looking at resources that are needed. Creating resources is a very hard task. It's easy—or easier—to create curricula. It's harder to get them adopted. We have tried not to prescribe things. Part of the reason for adoption is that we have not been prescriptive in our curriculum. We have given lots of options in many ways that one could adapt. I think that's part of why the community has come in—they see value for their own growth. There's an incentive for them to participate so they can intellectually contribute, develop resources, use resources, and move forward in their career.

We have also done some surveys. We try to continually take feedback from the community and see what are the hurdles and what are the barriers to adoption. There has been a continual effort to understand their needs.

Q: How were you able to get the workshops to be self-sustaining?

SP: We got the early adopters back to these workshops. That is another component which may be crucial—early adopter funds were also supplemented by travel funds to come back to the workshops. A lot of institutions don't have support for travel, so this way they get supported to either present a paper or present posters.

Now the workshops have become a very well-oiled machine. We have the PC chair, we have the PC vice chair, and an organizing committee and all of that, and the expectation is that the vice chair becomes a chair. The whole idea of bringing people in is that we tell them, "You're going to be a vice chair for the next conference, and your role is needed because we really want to reach out to newer communities. You bring new ideas, new energy, fresh blood." All of that is important to us. This is very explicit. We just need to reach out and hand things over to the community, so it starts rolling by itself.

Transparency is also important. We really want to show what we are doing and that nothing is hidden from anyone in the review process or in the submission process. Anything that gets accepted is by committee, not by one person. I think openness is important because I have seen a lot of communities become cliques, and it's very difficult to break into those cliques. And so with the new program chairs, they define what the next workshops' agenda should be and the previous leadership is only in an advisory role.

Q: What has been surprising or easier than you expected?

SP: What was surprising is the way the community rallied around this, and we still have this teleconference every Friday after all these years. It's amazing that we still have things to talk about. Normally, people get bored, they want to move on, but clearly this is something that is a mission for everybody. Was it easy? If you have people with you, then it becomes easy. And so in that sense, it has been easy.

I think in the beginning we were actually concerned that even to create a curriculum, there's so many things that everybody wants and how do you make them agree on the essence of what it should be? I would say that was the main bottleneck. But once you start off and then you get good people, it just happens. It's all about getting very interested, dedicated folks. And then it's also about shared vision. I've been fortunate in that way.

Q: What advice would you give to someone with a great idea and who wants to have a big impact?

SP: It's important to actually go with what is important for the community, what their needs are. I think that's important. And the community has driven it. Maybe you have some vision, but if there is a community need behind it and the community buys in, then it becomes more successful.

Q: What does success look like?

SP: In the end, I think the effort really should get so much adopted and so much ingrained in the culture and the educational ecosystem, that the effort that we are engaged in should become irrelevant. If you have adoption then the community takes over and we will basically put ourselves out of business. We want to be out of business. I mean that's really the goal.

References

[1] Bort, H., Bunde, D.P., Butler, Z., Hovey, C.L., Spacco, J. and Taylor, C. 2020. Conversation with a prominent propagator: Leo Porter. *ACM Inroads*. 11, 1 (2020), 12–15. DOI:https://doi.org/10.1145/3381023.

[2] CSinParallel: Parallel Computing in the Computer Science Curriculum: *https://csinparallel.org/*. Accessed: 2020-03-31.

[3] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery.

[4] Prasad, S.K. et al. 2012. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing -- Core Topics for Undergraduates. https://tcpp.cs.gsu.edu/curriculum/. Accessed: 2020-07-14.

[5] Prasad, S.K., Gupta, A., Rosenberg, A. and Sussman, A. eds. 2018. *Topics in Parallel and Distributed Computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms*. Springer.

[6] Prasad, S.K., Gupta, A., Rosenberg, A.L., Sussman, A. and Weems, C.C. eds. 2015. *Topics in Parallel and Distributed Computing: Introducing Concurrency in Undergraduate Courses*. Elsevier Science & Technology.

[7] Stanford, C., Cole, R., Froyd, J., Henderson, C., Friedrichsen, D. and Khatri, R. 2017. Analysis of Propagation Plans in NSF-Funded Education Development Projects. *Journal of Science Education and Technology*. 26, 4 (Aug. 2017), 418–437. DOI:https://doi.org/10.1007/s10956-017-9689-x.

[8] Taylor, C., Spacco, J., Bunde, D.P., Butler, Z., Bort, H., Hovey, C.L., Maiorana, F. and Zeume, T. 2018. Propagating the Adoption of CS Educational Innovations. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (July 2-4, 2018, Larnaca, Cyprus. New York, NY, USA, 2018), 217–235.

David P. Bunde
Knox College
Galesburg, Illinois, USA
dbunde@knox.edu

Zack Butler
Rochester Institute of Technology
Rochester, NY, USA
zjb@cs.rit.edu

Christopher L. Hovey
University of Colorado Boulder
Boulder, CO, USA
hoveyc@colorado.edu

Cynthia Taylor

Oberlin College

Oberlin OH, USA
ctaylor@oberlin.edu