Conversation with a Prominent Propagator: Paul Tymann

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

Improving undergraduate education in computer science requires that innovative teaching practices are scaled beyond the original developers, so that they become an integral part of teaching. However, achieving widespread adoption requires considerable time, energy, and planning. CS educators and education researchers must engage in intentional, sustained efforts to explore and implement successful approaches for increasing the awareness and impactful usage of transformative teaching. To that end, this column represents the next segment of our efforts to share knowledge about successful propagation strategies within the CS education community by interviewing *prominent propagators*, individuals who have successfully spread educational innovations [1–3].

In this column, we interview Paul Tymann. Paul is a Program Director in the Division of Undergraduate Education (DUE) at the US National Science Foundation (NSF). Prior to his current, permanent appointment at the NSF, Paul was a Professor and Chair of the CS department at the Rochester Institute of Technology and served a three-year rotating term as a Program Director at the NSF. He has also been the Symposium co-chair and Program co-chair of the SIGCSE Technical Symposium, the Chief Reader for the AP CS Principles exam, and a member of the ACM Education Council.

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

All information provided in this interview represents the opinions of an individual Program Director. The only official source for NSF policy is published materials that can be viewed on their website [7].

Q: How did you get into CS education research?

PT: My original research was on the system side of the house. I was doing networking and then parallel and distributed computing for a while. But I realized what I really enjoyed doing was teaching. And I think as any good teacher, I wanted to be able to improve the way that I taught and do it more effectively for students. And that just kind of naturally led to an interest in CS Ed research. So that's the pre-NSF story.

I had served on NSF review panels and things like that, but I was really curious what went on inside the building. I wanted to see how the sausage was made, so to speak. And I thought I would get a real vision of what was going on in the country as far as CS Ed research was concerned because most of the funding probably comes through the division that I'm in. Later the opportunity came up for a permanent position, and I was encouraged to apply. So I did, and poof! Here I am.

Q: Generally speaking, how does the NSF think about the propagation of research?

PT: It's obviously really important to the NSF. Whenever I recommend a proposal for an award, I'm making an investment, right? I'm literally taking taxpayer dollars and placing a bet that whatever this project is, whatever these PIs are going to do, is going to be successful. And, like any other investment, I want to see a return on that investment. Obviously an award at one particular institution benefits that institution, but a much better investment is one that propagates—although I think, you know, in NSF parlance I would say "disseminates." We want a project that has a broader impact, which basically means it gets propagated. It's used and it changes the community—hopefully for the better.

Q: What is your advice for faculty regarding how to maintain a project once it has been developed?

PT: One of the problems with NSF funding is that it's not long term. It's very short term. The longest awards that we typically give are five years. Most awards are around three years, and that money just basically gets things up off the ground. That is an interesting problem.

I have been thinking about some of the projects that have been more successful in terms of propagation and dissemination, and I really think it's projects where there's social support. For example, there's H-FOSS (The Humanitarian Free and Open Source Software [5]): that's a pretty vibrant community and people are contributing things on their own, and it seems to me, those are the sorts of efforts that work out the best.

Q: Do you have any advice on how to build a community around your project?

PT: From the very beginning, you should be thinking about how you are going to build a community. That needs to be a basic, fundamental part of what you're doing. How do you keep this from being your baby, or something so specific that it turns into one of these projects that just really doesn't make sense for others to pick up? When I think about the successful projects that we can all name, I think the common thread that runs through all of them is that there's a strong user community. I look at Media Computation, BlueJ, H-FOSS—in all of those, there's that very strong community.

But, you have to get your own ducks in order first, before you try to convince other people that the way you're quacking is the right way to quack. I think if you take a look at a program like IUSE, it's designed to do that. You have the level one projects, which are exploratory, and then when you go into the level twos and threes, you're taking something that worked and you're trying to propagate it. You can get more people to use it.

Q: Are there other strategies that people can employ to help others either take it up in the first place or continue to build on a project like that?

PT: One thing that we don't have in the CS Ed community that I think would help with propagation is some sort of a repository. If you take a look at open source software, they have github, right? There is

infrastructure and a kind of social ethos, for lack of a better word, where people participate and contribute. People have tried to make a CS Ed repository before. The NSDL (National Science Digital Library [6]) was a digital library that was started years and years ago, and NSF tried to mandate that people put things in the digital library—that was well before my time. The NSDL still exists but it is not widely used by the CS Ed research community.

I think part of the problem with trying to propagate is that there are several aspects to it. You certainly know one, which is maintenance. Another aspect is just getting it out. How do people know about it? I think that's big. The other thing that gets in the way is how much effort it takes a teacher to actually pick something up.

Q: If I am creating something as a researcher, what should I do to make sure it's easy for other people to be able to pick it up and use it in the classroom?

PT: Put out all the material that someone would need to actually implement it, and put it in a form that can be easily adapted. I think very rarely do we come up with absolute turnkey solutions. We all teach a little bit differently and I think an important part of this is being able to adapt materials easily. I would say that's one thing.

Some of the best dissemination plans that I see are what I call active dissemination plans. A lot of people will say things like, "We'll build a website and people will get it." That's the Field of Dreams approach, right? “If you build it…” but how do we know they're going to come? But a workshop—you can actually try to get people to invest. You bring them in. You show them what it is that you've developed, and then they can also help you with the development effort. I think if they get some skin in the game, in terms of that development effort, that helps a little bit with that maintenance aspect of it.

Q: How much active dissemination do you see in proposals?

PT: I think it's often the case that when someone writes a proposal, things like the evaluation plan and the dissemination plan are the things we often do at the end when the submission deadline is looming. I think proposers should think more about how they're going to propagate their work from the beginning. Research is good to learn new things, but it doesn't really have much of an impact unless you can actually get it out there. In terms of the research plan, I think an integral part of the project has to be, “How do I get that out there?”

Q: How could we improve how people find out about innovations in CS education?

PT: A lot of projects do good things at schools, but people really don't hear about it unless they happen to go to the one presentation at SIGCSE where it gets talked about. Sometimes it's just a matter of knowing that these things exist, and maybe we could have a column in Inroads that highlights a project or two, or three, every issue. Maybe we need to do a better job as the CS Ed research community in telling people what's out there, instead of assuming that they're going to go in and paw through the literature.

I think one of the things that you need to factor into that is how much effort does it take to find. If I'm scrambling to teach a new course, I don't know if I really have the time to do any kind of a literature search for that. It might be nice to have a place where I can go that might have some of these projects listed so that I could see them maybe say, "Hey, you know, that looks like that might help me here." And maybe that would help.

Q: What do you see from proposers in terms of working with users and adaptation?

PT: As you can imagine, you see things all over the spectrum. I think active dissemination resonates better with reviewers. It gives the reviewers a better feeling that this is something that may actually get picked up. As a result, those proposals get better reviews and so they're more likely to be funded. I don't want to say the vast majority of the proposals have these really well thought-out, active dissemination plans, but there is a significant number that do, and I think they tend to get rated better.

Q: Are there specific things in people's active dissemination plans that have stood out as being particularly good?

PT: One thing that I've seen people do is work with a publisher to actually publish the work. The Media Computation work that Mark [Guzdial] did was NSF funded, and a book came out of that [4]. I think that the book makes it much easier for someone to pick up. And, now you've got a publisher with some skin in the game who's going to help to try to maintain that.

Q: How do you measure successful propagation?

PT: When something is really successful, most people kind of know about it, but I do think collecting metrics is important. There are some projects that seem to get stuck—for some reason they don't propagate beyond the local or regional space. In your proposal, I think trying to explain how you're going to get this thing adopted is important. You're not going to have the metrics until after the project's done, but from my perspective, putting together a plan that will convince me that this stuff will get propagated is important.

Q: What causes projects to fail to propagate?

PT: I think a key part of all of this is somehow getting the information out. It's something that I don't think we do very well as a community. Also, sometimes there have been cases where people have done things that are so specific to their particular environment that any attempt to change it tends to break it.

Q: Have you seen projects that surprised you in terms of how they were picked up?

PT: To be honest, I don't know if I have been involved with NSF long enough to be able to answer that question. One of the things that I'm finding interesting now in my second time at the NSF is some of the

projects that I recommended my first time are actually starting to come to conclusion. Some things I think surprised me in the opposite direction; there were things that I really thought would get picked up and make a big difference, but for some reason they have not been picked up as much as I thought they would.

Q: Any guesses as to why that might be?

PT: There's probably a higher threshold to change that you have to do on some of these things, where you have to make major modifications to the way that you might teach normally or have to make a significant investment of time and resources to adopt an assignment or two. As an example, and we are probably over it now, but in the beginning there was a lot of resistance to active learning. I think the resistance wasn't so much that active learning is hard to do in a classroom. I think it's just that it's very different and it takes a significant investment of time and effort to implement it correctly.

It seems to me that projects involving a fundamental change in the way that someone teaches are harder to get out there. Maybe those are the ones that need more workshops or more TLC [tender loving care] to get adopted, unlike the quick and easy things that you can just plop into a classroom.

Q: How do you see the relationship between "Broader Impacts" and broadening participation in computing?

PT: To be honest, I kind of view them differently. When the NSF started to push broader impacts, everybody thought about diversity and things along those lines, but to me, broader impact says that somebody else is going to pick this up, that we're investing money in some sort of project that is going to spill out and be adopted by other institutions. That said, I think that if you can show that the impacts of your project are going to affect particular groups in a very positive way, that makes your impact stronger. So, they are related, but to me, broader impacts are really that other people pick it up, and broadening participation is how we change the way that we teach to be more equitable—and it's more than just the way we teach. There's so much stuff that I think is endemic in our society, and we tend not to see the things that groups who are unlike us are experiencing. We need to make ourselves more aware of it. An important thing we need to do to move everything forward is to figure out how to teach more equitably across the board.

If I take a look at my experience going through college 40 years ago, the curriculum, pedagogy, and all those sorts of things are all still basically the same. There is a sage on the stage lecturing about programming, then data structures, then hardware, then theory, you certainly know the drill. Today's world is a very different place—the people that we're teaching are very different and they're coming from very different backgrounds. We just haven't picked up on that yet and I think it's important that we do. If you're broadening participation, your project is going to have a more significant broader impact. So maybe they do go hand in hand.

Q: How do you see propagation efforts in relation to research activities?

PT: I think there's definitely a feedback loop there because the more you can propagate, the more people who adopt whatever it is that you're developing, the more information you're going to get about the efficacy of using these particular things. I think about it like any class that you teach—the first time you teach it, it's always a disaster. And then you teach, you make observations, you collect information, you refine it. I think CS Ed research probably has a similar feel: you put things together, you put them out there, and then you want to start collecting information about how well they work and make them better. So, I think you could argue that propagation could be part of that research.

Q: What's been the most rewarding aspect of being at the NSF?

PT: It's the people. And when I say it's the people, it's both the people that I work with in the building, and it's working with PIs. I really enjoy trying to get an understanding of the big picture. The other thing is I feel that I'm in a position where I might be able to make a difference. Not that you don't make a difference in the classroom, but here, you could help get a project underway that, if it propagates, could significantly influence tens of thousands of students across the country. The NSF is a pretty cool place to be affiliated with. We are always looking for good program directors!

[1] Bunde, D.P., Butler, Z., Hovey, C.L. and Taylor, C. 2021. CONVERSATIONS: Conversation with a prominent propagator: Colleen Lewis. *ACM Inroads*. 12, 1 (2021), 15. DOI:https://doi.org/10.1145/3446779.

[2] Bunde, D.P., Butler, Z., Hovey, C.L. and Taylor, C. 2020. CONVERSATIONS: Conversation with a prominent propagator: Michael Kölling. *ACM Inroads*. 11, 4 (2020), 6–8. DOI:https://doi.org/10.1145/3428677.

[3] Bunde, D.P., Butler, Z., Hovey, C.L. and Taylor, C. 2020. CONVERSATIONS: Conversation with a prominent propagator: Sushil Prasad. *ACM Inroads*. 11, 3 (2020), 22–24. DOI:https://doi.org/10.1145/3410472.

[4] Guzdial, M. and Ericson, B. 2016. *Introduction to computing and programming in Python: a multimedia approach*. Pearson.

[5] Humanitarian FOSS Project: Building Free Open Source Software for Society: *http://www.hfoss.org/*. Accessed: 2020-10-20.

[6] National Science Digital Library (NSDL): *https://nsdl.oercommons.org/*. Accessed: 2020-10-20.

[7] National Science Foundation (NSF): *https://nsf.gov/*. Accessed: 2020-11-12.

David P. Bunde
Knox College
2 E. South St
Galesburg, Illinois 61401 USA
dbunde@knox.edu

Zack Butler
Rochester Institute of Technology
Rochester, NY 14623 USA
zjb@cs.rit.edu

Christopher L. Hovey
University of Colorado Boulder

1045 18th Street, UCB 315
Boulder, CO 80309
hoveyc@colorado.edu

Cynthia Taylor
Oberlin College
10 N Professor St
Oberlin OH, 44074
ctaylor@oberlin.edu