Conversations with a Prominent Propagator: Mark Guzdial

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

Advancing Computer Science (CS) undergraduate education requires increasing the adoption and impactful use of innovative teaching practices. This article presents the next in our series of interviews with *prominent propagators*, educators and researchers who have successfully increased the user base of pedagogical and/or curricular innovations [1–3]. Our intent is to capture and share their knowledge and experiences, so that others can plan and sustain successful ways to spread their own ideas.

In this episode, we present an interview with Mark Guzdial, Professor of Electrical Engineering & Computer Science at the University of Michigan. Mark Guzdial is best known for developing Media Computation (MediaComp) [6], in which students learn Python or Java while working on a series of media-focused projects including image processing and sound manipulation. He has also worked on the Runestone free ebooks platform [9], which provides Computer Science textbooks that include educational best practices. He is currently working on building computational supports for Mathematics, English, Language Arts, and History.

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

Q: What was the Media Computation development process like?

MG: It was really a participatory design process. I had a team of advisors who were faculty from the liberal arts and sciences who would review what I was doing and say, “Yeah, I like it. I don't like it.” Because who am I to pick what their students should learn about computer science? And then we did focus groups with students, as we're developing things saying, “Do you like this project? Do you not like this project?” For majors, we can say, okay, we are experts in computer science. We know what's important in computer science. But for non-majors where computer science is more of a literacy, knowing what they want to do with it is pretty important.

Q: Were you thinking about expanding it beyond Georgia Tech at that point?

MG: When we first started out with Media Computation, we looked and there weren't a lot of good liberal-arts-oriented CS courses out there for non-majors. So we did start it saying maybe we're coming up with a solution that people other than Georgia Tech would want to use. So that was an explicit part of what we wanted to have happen, but I figured we weren't going to get there if we didn't make it succeed at Georgia Tech.

2153-2184/2022/March $15.00

http://dx.doi.org/10.1145/3497877

I was familiar with situated learning from Lave and Wenger's book [8]. It's a theory of learning which basically says that everybody is learning in order to become part of a community of practice. First you engage in what's called a legitimate peripheral participation. You work in the mail room and things like that, which gives you a chance to find out how the place works. So the first thing that we should do if we are building Computer Science for liberal arts majors is ask, "What do liberal arts professionals do with Computer Science?" This was 2002, so how many history professors are using programming in their daily practice? We actually wrote a paper about how we were creating illegitimate peripheral participation. With MediaComp, we said, “What might a world look like where liberal arts professionals use programming regularly?” and we then tried to create a course to make that legitimate.

Q: What other projects are you doing now that Media Computation has been so successful?

MG: We’re looking at what else LSA (Literature, Science and the Arts) students need. We've got dozens of interviews and a hundred faculty surveys. We've come up with three themes for Computing in LSA. There is Computing for Discovery, which is a little bit different from data science. Computation for Expression—this is where Media Computation fits in, but also they have faculty now who are doing things in VR and AR, extended reality, and installation art. And the third one is Computation for Justice: looking at our systems and asking, where are they inequitable? Where are they making inequalities which exist in our society worse? When we did our survey, 49% of the faculty said that all students should be required to learn about Computing for Justice. Can you imagine 49% of faculty at any large college agreeing that anything should be required for everybody? I was pretty blown away.

Q: How do you get other faculty on board to want to teach that content, especially if they're out of the department?

MG: For LSA, they recognize that computing is becoming increasingly important, even for the students who don't want to major in it. And so they recognize that they have a need, but they didn't really know what those needs were. And now comes the bigger questions. How do you meet those needs? Courses are pretty full. Are you going to make new courses? Are you going to try to adapt existing courses? So coming up with an implementation plan is next. But there's a third driver, which is the economic needs of the college.

Q: How has your relationship to propagation changed as you’ve done different projects?

MG: I originally thought that my challenge was to produce data—let me prove to you that students are learning. Then Lijun Ni started looking at who adopted and why, and in the end, what mattered was the teacher's excitement. Full-stop.

This is my favorite story: The head of Computer Science at a university desperately wanted to do MediaComp, but they had one tenured faculty member who taught all of their CS 1 and CS 2 and he refused. So the head did an experiment with him, saying “I'm going to teach a CS 1 with MediaComp, you go teach whatever you want. And then, we'll look at their final exam in CS 2 for all of the students who go on and see if you can tell the difference.” It worked. In the normal class, there was about a 50% withdrawal and failure rate. In the MediaComp class, it was 25%. There was no difference between the

students on the final exams. So, this faculty member said, “Okay, I'll teach MediaComp.” For the next year, he was my absolutely worst MediaComp adopter. He wrote me weekly to tell me how terrible the curriculum was, how everything wasn't working. After a year, he announced, “Okay, I've tried it. I won't do it again.”
We can characterize his perspective as “My job is to produce the best and brightest possible software engineers. I don't see media as a way of doing it. I don't like media. I am not interested in improving retention.” So in the end, I realized that dissemination is really an affective motivational issue, it isn't anything to do with rational purposes. I did work with David Fossati, where we did a bunch of interviews to try to explore that [5]. We found research studies did not have a big impact, and people often looked for reasons not to adopt something now. “Well, that was at Georgia Tech. We're not like Georgia Tech. It wouldn't work here.”

Q: How did that experience change how you see propagation?

MG: I became a big fan of the Increase the Impact work by Jeff Froyd and Charles Henderson. The way that you get dissemination and propagation is you start out from what their problems are. It's really a marketing kind of thing: what are the pain points, and how do I solve your pain points? I think the eBooks have been successful because it matters that they are free. It matters that there's all kinds of activities.

Now, the fact that we're trying to build things on top of educational psychology principles, that we're trying to be aware of cognitive load, that we are trying to make a lot of worked examples—I don't think it helps us a lot with adoption of the eBooks, but can we get people who adopt these eBooks to see the underlying practices that emphasize active learning or worked examples? And can we help teachers to adopt those practices? For Barbara Ericson, my wife and research partner, that is explicitly her goal.

Q: How has this impacted your current work?

MG: I'm doing work now with trying to build task specific programming languages that meet teachers' needs in other disciplines. I work with history teachers. I have this terrific collaborator, Tammy Schreiner; she's a professor in history at Grand Valley State. She says people don't become history teachers because they love numbers and computers. But social studies standards in most of the states expect teachers to teach about data visualizations, graphs, charts, timelines, and maps. Part of it is figuring out the pain point: if you've been able to ignore data visualizations your whole career, how do I convince you? Well, they are in the standards. Let's see if we can figure out a way for you to teach them.

A lot of what we do is participatory design. We put multiple visualization tools in front of teachers, and ask them to play with them and give us feedback. Usually our tools are just in the mix, it's not the only thing they're looking at. And then we'll get from teachers, “Oh, I see what you're trying to do here. Yeah. That really sucks. Let me tell you how you can make it good” and it's great. That's terrific feedback for our process, but now we're at a point where teachers are saying “Your tool is the best one of these here. I really like your tool,” but nobody's adopting yet. So we're now to the point where the tool meets their criteria, at least among the ones that they'll tell us. Are there tacit criteria? Are there other things going

on? That's where the research is going now, to try to understand the other factors that influence adoption of this kind of technology. Our whole project is about how we get to adoption in non-STEM disciplines and social studies with explicitly computing activities. That's a hard one.

Q: You've had experience both trying to get postsecondary educators to adopt your things and also trying to get K-12 educators to adopt things. What are the differences between those groups?

MG: One of the factors that I mostly know from research on the K-12 side, which I've never heard as being a major factor of the Higher Ed side, is the context in which the practicing teacher is enmeshed. One of the great findings of Anne Leftwich is that pre-service teacher education doesn't have a lot of impact on practice, because the biggest factor on practice is what the teachers in their school do [10]. When you're a teacher doing student teaching, you're going to do what your model teacher is doing, not what you were taught. Then when you go get a job, you're going to do what's accepted there. Some of it is: what technology do you have available? What are the standards they've decided to focus on? What are the tests that the students in that school take? Context is everything in terms of teacher practice. I don't know if that's as true at the Higher Ed level. For the most part, Higher Ed teachers don't see each other teach. So it's not clear that we know a lot about the context. It certainly influences us in different ways.

Q: What differences do you see within Higher Ed? Are there institutional or programmatic differences that have influenced your strategies for encouraging adoption?

MG: At Georgia Tech, I was in a College of Computing and the way that they thought about CS education is very different from how they think about CS education in a College of Engineering. It's going to sound like a tautology, but a College of Engineering produces engineers—practicing professional engineers. They offer almost no service classes. At Georgia Tech, we taught all of the non-majors Computer Science. We had degree programs like a BS in Computational Media and the Human-Centered Computing PhD. We had this strong sense that computing is for lots of people, not just future engineers. When Computer Science is in a College of Engineering, it's about producing engineers. So today, minors and non-majors can almost never get into the classes. That is a big difference because of what you value, what you're trying to achieve.

MediaComp has been successful where it met a need. I visited UMass Amherst a couple of years ago. I met with some of their CS faculty who teach with the IDE that I created for MediaComp, and that was really exciting. They don't teach MediaComp; they just thought it was a particularly nice Python editor. There's a lot of AP CS teachers who teach with the picture lab that Barbara created because it's there and kids like it. Has it changed their perception about who should do computing or that we should be doing computing for creative expression? I doubt it. I think for some of these things that it's like paradigm changes. I don't know how often you convince people to change identities.

Q: You mentioned that some people use your editor or ebooks because they need a textbook and it's free. How do you see adaptability and faculty's ability to use a piece of your different innovations?

MG: If you're using JES because it's a fun Python IDE, do you maybe throw in some media because it's already built in and the help is right there? It's already available [7]. Does it become a slippery slope to try to encourage other kinds of behavior? One of the things that Barbara is trying to do is encourage Peer Instruction by taking multiple choice questions out of the ebook that people have already been using. That's such a lovely, slippery slope: you're already using the ebook, so why don't you do it as a Peer Instruction question?

I'm a big fan of behavioral economics. Removing barriers to innovation is easier than convincing people to adopt an innovation. That's what we're trying to do with "Why don't you try the IDE? Why don't you try the ebook? Okay. It's pretty cool, huh? Well, by the way, you can now also do this other stuff."

This is purely a hypothesis, but there are some innovations which are generative in that way. They create these opportunities to change mindsets, to change practices. That's sort of another form of these questions you're asking: what are the characteristics of an innovation that might lead to greater changes in practices, attitudes, and other innovations later?

Q: What is successful propagation? How do you know if you've achieved it?

MG: My definition has definitely changed over the years. When I was doing MediaComp, one of the things I realized was that writing the books was key. I talked to so many people who said, "Well, I saw your book and so I decided to try it." Things like lesson plans, things like worked examples, these matter if you want people to adopt. Books really matter. I remember at one point when my publisher contacted me and said my book was currently the third most adopted Python textbook in the country, that was really exciting for me. But now, especially with the work that I'm doing with K-12, I've got a half dozen tools for doing data visualization in a social studies class; if I can get a whole bunch of social studies classes to adopt one of them, I would consider that to be a huge win.

Q: What advice do you have for somebody just getting started?

MG: I don't know whether I learned this from the Increase the Impact work or from Betsy DiSalvo's participatory design work [4], but the big thing is that you first have to figure out who's going to want this. Most design does not work like the iPhone, you know, "I've thought of a need that nobody else had. People don't even know they have this need yet." That just doesn't happen often. People are rational beings and their classes are packed. They believe that what they're teaching is important. You want to convince them that your thing is more important. That's hard to do. Show them that you can solve their problems so they can teach something more successfully or that they can teach something that they haven't been able to do previously. Or you can save them time. I think that is much more powerful.

References

[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), 15–17. DOI:https://doi.org/10.1145/3446779.

[2] Bunde, D.P., Butler, Z., Hovey, C.L. and Taylor, C. 2021. CONVERSATIONS: Conversation with a

prominent propagator: Paul Tymann. *ACM Inroads*. 12, 2 (2021), 10–13. DOI:https://doi.org/10.1145/3446392.
[3] Bunde, D.P., Butler, Z., Hovey, C.L. and Taylor, C. 2021. CONVERSATIONS: Conversation with a prominent propagator: Tim Bell. *ACM Inroads*. 12, 3 (2021), 14–17. DOI:https://doi.org/10.1145/3457774.
[4] DiSalvo, B., Yip, J., Bonsignore, E. and Carl, D. 2017. Participatory Design for Learning. *Participatory Design for Learning*. Routledge. 256.
[5] Fossati, D. and Guzdial, M. 2011. The Use of Evidence in the Change Making Process of Computer Science Educators. *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2011), 685–690.
[6] Guzdial, M. 2013. Exploring hypotheses about media computation. *Proceedings of the ninth annual international ACM conference on International computing education research* (New York, NY, USA, 2013), 19–26.
[7] JES 6.0 Final Release: *https://github.com/gatech-csl/jes/releases*. Accessed: 2021-11-09.
[8] Lave, J. and Wenger, E. 1991. *Situated learning: legitimate peripheral participation*. Cambridge University Press.
[9] Runestone Interactive Community and Project: *https://runestone.academy/*. Accessed: 2021-09-21.
[10] Tondeur, J., van Braak, J., Ertmer, P.A. and Ottenbreit-Leftwich, A. 2017. Understanding the relationship between teachers' pedagogical beliefs and technology use in education: a systematic review of qualitative evidence. *Educational Technology Research and Development*. 65, 3 (2017), 555–575. DOI:https://doi.org/10.1007/s11423-016-9481-2.

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