Conversation with a Prominent Propagator: Tim Bell

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

Improving computer science (CS) undergraduate education requires sustained, intentional efforts to increase the adoption and meaningful use of innovative teaching practices. This article is part of a series of interviews [4–6] with *prominent propagators*, people who have successfully spread innovations in CS pedagogy or curricula. Our goal is to capture their experiences so that others can learn successful techniques and insights for spreading their own ideas.

In this installment, we interview Tim Bell, Professor of Computer Science at the University of Canterbury. Tim is best known for developing CSUnplugged [7], a collection of free resources for teaching CS concepts that do not require a computer. Initially developed for pre-college students, the materials are now widely used for a variety of audiences, including CS majors, non-majors, and community outreach in many countries [1–3].

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

Q: How did CSUnplugged get started, and how has it evolved over the years?

TB: Originally, it was designed to be used by an academic going into a classroom to enable them to say, “This is what I do for a living” in a meaningful way. The material was written for a computer scientist who didn't really need a lot of explanation of what the point of it was. It's since been picked up in elementary school classes that are often taught by people who don't have a CS background. So the material now explains a lot about why we care about these topics, and how they affect people in everyday life, as well as taking educators through exactly what to do and exactly what to watch out for and so on.

I would say that primary school is still the biggest target. I use it for my university classes and certainly for community talks too. People find it quite valuable because the public misunderstands CS. Having parents understand is really important. Also politicians and administrators, because a lot of them might first think, “What are the financial implications?” but after five or ten minutes using Unplugged to experience CS, they’ll say, “This is nothing like what I thought it was. There's some substance to this subject.” I’ve also enjoyed using it with senior citizens.

2153-2184/2021/September
http://dx.doi.org/10.1145/3457774

Q: When did you switch from developing exercises to thinking about how to get other people to use them?

TB: I think it's always gone hand-in-hand. Mike Fellows has worked very closely with me on this and we met purely by chance on the internet, around 1992. He had a couple of projects about getting maths into the community. One of his goals was changing the school system, because it was also the time when computers were coming in. He referred to schools having a cargo cult attitude—the computer turns up in the classroom and students do really uninteresting stuff on it, but it must be good because it's on a computer! He was interested in making it more meaningful for kids but wanted to change the education system as well. So it's always been about communicating to people.

Q: How did you initially get teachers to use CS Unplugged in their classrooms?

TB: The one thing that has really taken us by surprise is that it has developed a life of its own, partly by chance and partly because we've always wanted to make it easy to access. Initially when Mike and I got together, we built up the whole canon of ideas and put it together as a book for computer scientists, and tried to get it published. It was rejected many, many times because no one could pigeonhole it: the computer department said that no one's going to use it on a computer and education editors said that it's about computers so we should talk to their computer department.

In fact, the rejections were the best thing that happened. I don't think Creative Commons existed back in the early '90s, but we made it shareware. The web was just turning up, and people weren't that aware of CS Unplugged except in CS departments. Through the '90s, a following formed and people started using it primarily for outreach to go into schools. Mike and I would use it when we were asked to do something in a school, but we otherwise didn't focus on it, since our main research areas were elsewhere.

Q: When did the ideas start gaining traction?

TB: In 2003, the ACM published the model K-12 CS curriculum [8], which was quite radical back then. One of the immediate questions was how can you possibly teach CS in K-12? A lot of the examples that they quoted were from our website, which kind of blindsided me. When it came out, within days I started getting emails from all around the world from people saying, "Tell us your philosophies and your teaching methods" and so on. And I was thinking, "It was just something that I was doing 10 years ago in my kid's school."

Because a lot of countries were thinking of introducing elementary level computer science, I ended up being invited to go and speak overseas at a lot of these places. I said, "Look, this is just something I've been doing privately. I'm not an education expert." But I was invited to events on the foundation of curriculum in a whole lot of different countries. So I'd sit there hearing all the other speakers and it would all start to resonate and I'd think, "This is a really good way to adjust things, and this is what people really need.” We'd adjust the way that we'd present and from there it snowballed. By the time I was going to my 10th overseas conference, I was able to say, “Well, as happens in these other countries...” and people go, “Whoa, you know so much about this stuff!” Really, Unplugged was just my golden ticket that got me in on the ground floor of a lot of these discussions. As a result, CS education is now my main research interest.

Q: What else did you do to promote it?

TB: When I finished being head of department, around 2005, I was given sabbatical leave. At the time, my main research area was computers and music. I got this fellowship that enabled me to go to a lot of universities around the world. I offered to give a seminar and suggested three titles: a couple of them were about music and one of them was about Unplugged. Nearly everyone said, “We want to hear about the Unplugged thing,” and it ended up turning into a kind of missionary journey.

I guess we've always had an approach where if anyone's interested in Unplugged, we'll talk about it. And if we can find a way to do it, we'll get there, often piggybacking off other travel. I'd be at SIGCSE presenting a paper about a different subject altogether, and we would run the workshop for next to no cost because we're already there.

The other thing about Unplugged is that, because it doesn't use technology, people can pick it up and use it straightaway. Years later, someone would say, “Oh, you told me about this, and I've been using it ever since.” I had no idea. I think that with a lot of projects in CS, programming languages or things like that, they gradually change. Eventually, they stop being supported, or they don’t work on particular versions of an operating system. That was never an issue for Unplugged. Once someone got enthused, it just carried on rather than needing a lot of care and maintenance.

Q: It seems like Unplugged is very easy for people to implement: how has that helped with propagation?

TB: Exactly! Sometimes when I've been talking to people, you just get a paper napkin, tear it up, draw some things on it and say, “Right now, we've done this concept from computer science. That's how simple the resources are.” So, for example, we've done a new version of it for people under lockdown, and we've deliberately done it in a way that we're not assuming you've got access to a stationery cupboard. We've tried to do it in a way that you can do it just using packaging and whatever's lying around, which says more anyway, because a lot of CS is about abstraction and we really want to get away from people going, “Oh, you must have a particular size piece of paper and a felt pen.” Like you say, that makes it very easy for people to adapt it to whatever they have locally.

Q: What have been some of the biggest challenges to getting people to adopt?

TB: We haven't really seen huge challenges. There have been times where people misunderstand it or called it a curriculum and we're pointing out, “No, it's not a curriculum. It's just a few ideas to use as part of your curriculum.”

The challenges that we've had are probably more around overenthusiasm. We encourage translations into other languages and adaptations so that things make more sense in a different culture. For example, Unplugged is very popular in South Korea for some reason, to the point where I've come across about six different translations of it. Each is different: one is web-based, and one is textbook-based, and one is more cartoony, and things like that. It's great having that diversity, but we almost get to the point where people are asking, “Why are you letting that other person translate it?” It gets a bit political. But we were really pleased that we've kept it as a Creative Commons thing that people can use it how they want. The other thing about being Creative Commons is that it has developed a life of its own now and it doesn't depend on particular people.

Q: How have you gotten other people involved to make activities?

TB: It's been a little haphazard. There are maybe a dozen people around the world who have really actively developed new activities, and some have been amazing and we have added them to the main canon. Other times, I look at something and think, “I don't know how well that would really work in general.” One of the principles is that you have to try it a few times with kids in a class, because often something that doesn't look great turns out to be amazing and vice versa. So we put a section on the website that we call the Community Section, which basically means “untested: might be absolutely amazing, don't know.” We want to share anything that people have come up with. It's a bit serendipitous when things happen and how well they happen.

Q: What adaptations or additions are you working on?

TB: One thing that's only half baked at the moment is a “where do I start?” guide. There's so many people who get told to use Unplugged and they end up on our website. At the moment it's not too overwhelming, but if we had 30 or 40 or 50 different activities, they might not know where to start. So it's become more important to guide new people through it than to develop new activities.

Adaptation tends to be more about updating and modernizing things—the original one was written back in the 90’s, and I remember modernizing it about 10 years ago. Pretty much the only things I had to change were terms like “floppy disk,” which became “USB drive.” I guess one reason it's had longevity is that the general principles are things like, “there are slow algorithms and fast algorithms,” and those are not going to change.

A significant adaptation that we’ve been working on is connecting Unplugged to programming. It’s becoming clear that Unplugged is a useful complementary tool to use in conjunction with teaching programming, so we’re working on ways to connect them explicitly.

Q: How do you determine when to make updates?

TB: The needs of the community and also funding guide us. Microsoft just contacted us out of the blue about five years ago and said they had noticed that many of the things they were funding around the world were using Unplugged. They were basically saying it would be better if we got the translation process online. Now we've got an online system where translation can be crowdsourced, and people can translate a paragraph at a time and someone else can approve it. So, it makes the translation process a lot more organized and doesn't rely on one person.

The most significant new version was where it went from being written for academics to being written for teachers. We did this with a grant in New Zealand from a science communication organization, and we paid a couple of teachers to translate it from academic language to teacher language.

Q: What does successful propagation look like?

TB: I think successful means that it's taken on a life of its own. And that happened a while ago. If I stopped doing it—and I'm really enjoying myself, so I'm not going to—but if I completely stopped doing everything overnight, it would still carry on as a thing. All the website material is open source, all the written materials are Creative Commons, and so on.

Q: What's been the most rewarding aspect of getting this out there into the world?

TB: As any teacher knows, it's that moment when a kid's eyes light up, but for me, my students are often teachers. I run classes for teachers and it's when their eyes light up and they go, “Well, this is really making sense. You've given me something that makes a difference for me.”

I guess the other one is when a sensible curriculum gets adopted. I think that it's a struggle in many countries, even when you think you've got an okay curriculum, someone will misunderstand the wording of it and turn it into something that it's not, or turn it into what they can teach rather than what it was intended to be. So it's rewarding when you see things being done properly, when the curriculum is about the Computer Science that actually makes a difference in the world, rather than just having fun with computers.

Q: What has surprised you?

TB: When we were first making it available to people and it was becoming popular, I thought if we got the time and resources, we would develop more activities. Although we've been gradually doing that, what I find is most people focus on just half a dozen of the activities and that's enough. Frankly, that probably is appropriate because there are lots of other things people should be doing. Learning programming takes time on task and that's a part of most curricula, whereas Unplugged is a quick way to get people in the door.

From that point of view, what's happened more—which I didn't expect—is that we're just adapting the same activities for many different situations: for different cultures, for working at home, for different grade levels—all sorts of different ways that it can be packaged, but the same basic activity.

Q: What advice would you give to someone who has something that they want other people to adopt?

TB: The approach that we've tended to take is a very open one, where we're saying, "These are just ideas. Someone else could have had them. We happened to write them down, but we wanted to make it as easy as possible for you to use it. And we're listening to your ideas about how you might change it to make it even better." So basically, offering it with open hands rather than putting it behind a wall is what has worked really well for us. I appreciate that for some people, if they're developing a product, that may or may not work. We've got the benefit of being in a university system. I've got a day job that will pay the bills and so on. It enables us to take risks and that would be the next bit of advice: just take risks.

References

[1] Bell, T. 2020. Teaching Teachers to Teach Computer Science - Unplugged or Plugged-in? *Proceedings of the 2020 ACM Conference on International Computing Education Research* (New York, NY, USA, Aug. 2020), 1.

[2] Bell, T., Rosamond, F. and Casey, N. 2012. Computer Science Unplugged and Related Projects in Math and Computer Science Popularization. *The Multivariate Algorithmic Revolution and Beyond: Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*. H.L. Bodlaender, R. Downey, F.V. Fomin, and D. Marx, eds. Springer Berlin Heidelberg. 398–456.

[3] Bell, T. and Vahrenhold, J. 2018. CS Unplugged—How Is It Used, and Does It Work? *Adventures Between Lower Bounds and Higher Altitudes: Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*. H.-J. Böckenhauer, D. Komm, and W. Unger, eds. Springer International Publishing. 497–521.

[4] 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.

[5] 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.

[6] 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). DOI:https://doi.org/10.1145/3446392.

[7] CS Unplugged: *https://csunplugged.org/en/*. Accessed: 2021-01-26.

[8] Tucker, A. 2003. *A Model Curriculum for K--12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee*. Association for Computing Machinery.

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