

Benjamin Carey: Music Technologist Artist Researcher

The surprise elements are very important for me so any kind of generative system or way of working using randomisation or algorithms that are kind of opaque is fascinating because when you stand away from that and interact with it as a performer, it provokes different ideas...

Benjamin Carey is a saxophonist, composer and technologist with interests in contemporary classical, improvised, interactive and electro-acoustic music. After completing a Bachelor of Music at the Sydney Conservatorium of Music in 2005, he moved to France to study saxophone and contemporary music under Marie-Bernadette Charrier at the Conservatoire de Bordeaux. Back in Australia, he completed his PhD which focused upon the design and development of interactive musical systems for improvised performance with instrumental musicians at the University of Technology, Sydney¹. He has performed and exhibited work in Australia, New Zealand, France, Austria, the United States and Switzerland and published a number of research papers².

Benjamin created *_derivations*, an interactive performance system designed for use in improvisatory musical performance. The system listens to the performance of its collaborators and uses this information to make decisions about how it responds to the unfolding musical dialogue. Ben and other musicians have performed live with the system demonstrating how these kinds of digital instruments can be used creatively. The interaction between human performers and digital system is often intricate and sometimes unpredictable, raising interesting questions about what makes a digital instrument rewarding to work with. Having ventured into the software world, his fascination with computational creativity has been focused on how this be designed to transform the way people behave. This interest springs from his own personal experience of being stimulated to do different things in musical performance by having a digital partner and what is more one whose responses cannot be easily predicted. It has moreover transformed his own creative process.

As a PhD researcher, Benjamin's approach to his work underwent much change and evolved into a deeply reflective process that is now an embedded part of his creative practice. He is fascinated by the potential of computational creativity because of its capability to prompt ("provoke") people to act in ways they might not by themselves or with others. There is a sense of excitement that comes through interacting with software you have created yourself, something that responds in a way you have not anticipated feels 'alive' as if partnering with another sentient being.

In his interview, Benjamin talks about the way research has influenced his reflective practice and the importance of surprise in parallel with musical coherence in his interaction with a self-created digital partner.



Figure 1: Album cover for *_derivations: human-machine improvisations* (2014, Integrated Records) ©Zoé Nelson-Carey (photo) and Holly Philip (design/layout)

Q: What is the practice- your creative practice?

B: For the vast majority of time, the creative practice is creating software. The end result, what the public sees is a performance and it might be a performance with myself with a piece of software, or it might be another musician interacting with that software and I'd be present or it could someone who has download the software elsewhere.

Q: What sort of tools do you use for that?

B: I use MaxMSP. I first used that to process my saxophone playing using Effects... Then I got more into it and started to think of my work with Max as creating larger systems to improvise with. Most of the practice is based in the computer. These days I am going more towards doing audio visual work but that's just one part of my work. I wanted it to be as unpredictable as possible so would feed new improvisations each time.

Q: How do you begin?

B: There is generally a seed for an idea so for instance if I am working on a piece of music and I want to use a process or an effect or some way of creating a musical composition or an interaction between myself and the computer that I have not encountered, I'll have an idea and it quite often comes from previous practice. Then I'll get into the software and start mocking up something, having not written down exactly what the idea was. Then it's sketching and I'll get into the software and sketch a little idea and usually that idea has some kind of input and output and that could be a microphone and that could be the mouse or something like that feeding off that initial process. Quite often the seed of the idea gets me to sit down at the

computer and work and then that process of sketching that idea usually forms this kind of feedback loop where I'm thinking 'OK the idea I had at the start has changed and is continuing to evolve and it is only afterwards I start thinking where am I going, is it any different to what I initially started out with? I can't usually put my finger on what it is I want to do until I sit down and do it. I design very much from the bottom up whatever it is, the building blocks, rather than having a broad overview of the software I want to design. That comes from my performer background. As a classical player, you need to get into small sections and you only put the large piece together at the end of the process.

Q: Is there a typical way of selecting how you want to move forward?

B: That's a really good question. It depends on the level at which I am working and how long I have been working on something but generally I will work and idea through until its eventual demise. Quite often I will get to a technical hurdle and I'll save that sketch and think about it later. Then I'll move into something else, get frustrated and go off and do other creative work. That process is generally stop start. If I am working on an idea, I'll really work it through until I realise that no that's going to take me another couple of days. I don't have the time now, I'll put it aside. So rather than sitting down and saying I'm going to achieve this in the next two hours and then add it to a larger system, it's very much in the initial stages, head down not really knowing what I'm doing and then taking a breath later on. You have a fresh look at it. You also draw connections between what it is you've done and other pieces of work. I keep all of my sketches so even if I don't do anything with them, a lot of these small patches become the seeds for other things. Quite often I'll look at something I knew what a dead end at one time but I'll be able to connect it to something else. In the process of actually making that sketch I don't see those connections because I am involved in the doing at that point. But definitely later on, I either see that it was a naïve idea and I couldn't take it anywhere else or I find that it connects to other ideas that I didn't realise.

Q: Would you describe that process as 'reflection-in-action'?

B: Yes, I'd see it as reflection-in-action. I found it a natural way of working. At first it was frustrating because I saw it as a natural consequence of being an inexperienced programmer. I have never been a computer scientist or had programming lessons. I learnt it all through doing tutorials and things like that. That way of working felt natural for me because it was 'tinkering' and at first if I got to a technical problem I would think that's because I didn't know enough so therefore I'd put it away but after a number of years doing it I found that that was the way I approached programming intuitively anyway. I think it's always been, since I started programming in about 2008, that's always been the way I'd worked and I didn't see it as reflection in action until I knew more about reflective practice.

Doing the PhD and writing the software I learned a lot about me as a creative practitioner because this was a completely new practice to me. I actually began programming two years before the PhD – in 2008. Before that I hadn't worked with computer software.

Q: Do you have any thoughts about how you judge or assess where you are?

B: I think if it works and there are no errors and I am quite happy with it and I start building it into something larger. Having said that the up and down in terms of lower level bubbling up, very small algorithms and higher-level things, I am always jumping down to the lower level stuff. If I get into a higher-level thing and I realise there's a functionality that's missing I have to jump down into that level. It's really difficult for me if it is a small seed that has come from a lower level, as long as that idea functions then I can take the next step. In terms of larger scale things, if it sounds good and if it either confirms my idea of what I wanted to achieve or surprises me in a good way, which is quite often what happens when you are programming, then I'll decide to keep it.

Q: In terms of the programming process do you have a motivation for challenging yourself or getting the unexpected?

B: Creating the `_derivations` software for my PhD thesis, it was finding ways of taking something that I knew, which was my saxophone performance, and presenting it in a way that could provoke me to do something different. If the computer can take something and draw connections and I don't understand what it's doing under the hood even as the person who has written it, then that's really exciting. The surprise elements are very important for me so any kind of generative system or way of working using randomisation or algorithms that are kind of opaque is fascinating because when you stand away from that and interact with it as a performer, it provokes different ideas- the performance is separate to writing the software. If you are interacting with something that is unpredictable it feels alive and that's thrilling as a performer and also as an interaction on stage. Also, for me interacting with something that is surprising gives you a different conception of what you want to create next. I created these random type systems and I'd perform with them and get quite excited but the more I spent time with them, I realised that I wanted them to be maybe a little bit less random or have more unpredictability but a sense of coherence, all of those kinds of things I didn't realise when I was first starting and it was the unpredictability that fed that I think. If I set an algorithm going and I am observing its output, quite often I'll add a simulated performance so if there is a pre-recorded performance of the saxophone that I've done, the testing of it is quite performative. I don't need to have my saxophone there all the time, I can just press play and have a listen it what it does. Quite often I see that as still part of the software engineering process; it's like injecting a past performance into it as a way of testing.

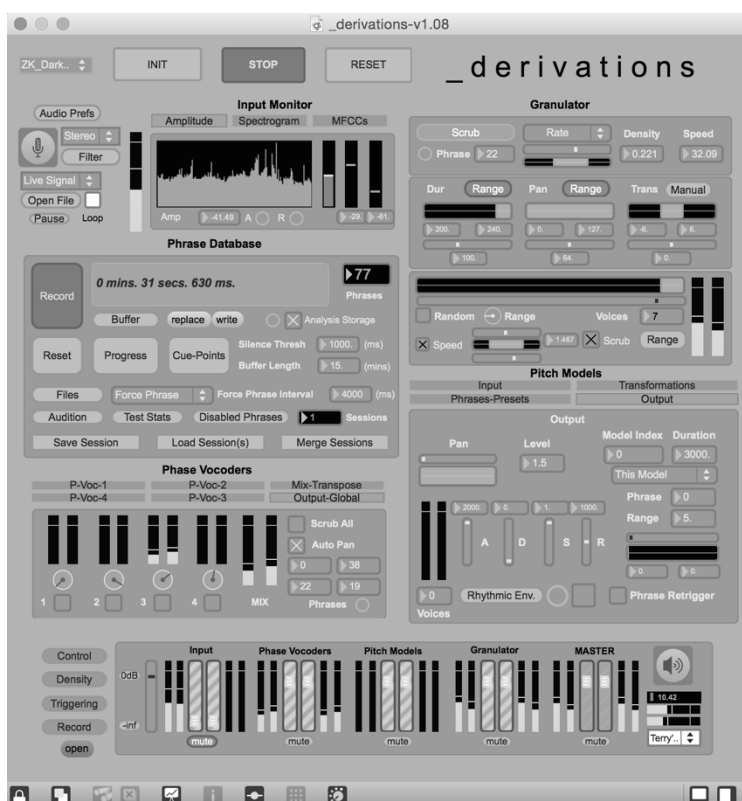


Image 2: “Screenshot of the `_derivations` software, version 1.08 released in 2015”

Q: What's the difference between a good surprise and a not so good surprise?

B: A really good question. A good surprise is when the software does something I didn't expect but is actually musical and is coherent in the context of my performance and everything that's come previously. It's quite difficult to judge because it's also in the context of improvisation so you don't want it to do the same every time.

Q: You don't want any predictability?

B: No exactly- but you also don't want it to go off on its own tangent and not be able to relate to things it's heard or to be able to provoke something that's in the style or context of what is going on now. If I'm testing it and a surprise happens, and then another surprise happens, and another and there's no consistency between the algorithm's output then it becomes random and it's difficult for me as the person who's assessing it from the outside to see if I can perform with that. So I play the role of the listener who created the algorithm. I'll test it and I'm thinking if I were in the audience, would that make any sense? When I interact with it, I am listening and responding to it and if it makes a decision that doesn't make any sense at all from what I've just given it, then it's not working and I can't connect that surprise to any musical idea that I'd like to pursue.

I have a general idea of what I want, but I allow that testing process to be the arbiter of whether or not something works, in order to find out what it is that I want. I have this broad idea but I can't tell exactly what it is until I test it. The testing process takes a very long time and it's very much embedded in the writing of the software. I find that after while it's only after I've stepped away from it and I've listening to recordings of that interaction a few times that I realise what it is that makes this thing tick and where to go next. It is very difficult to map it out from the outset.

There's been a number of performances when I've been playing with the software and it will generate something that doesn't conform to the interaction that we've been having. So it will bring something from its database that doesn't relate at all to what we've been having. It's frustrating but it's a performance so I need to try and rein it in in some way so I might start playing differently. I might try and provoke it to do different things. For instance, if I've been playing quite quietly, and interacting with the system in that way and the system has been quite quiet, and we seem to be in some kind of coherent dialogue and then it spits out something unexpected and seems out of context, there's one of two things I can do: I can try and play even quieter and move into a different area or I can start playing as if it has provoked me to play louder. So I'll go off with it and after I might go back to the software and think why did it do that? What is it about the internal algorithm that's happening there? Was it something to do with the fact that my microphone was too close to the speakers and it was some kind of interference? And then I'll drill down into the code and start thinking about mitigating those things.

And that happened a number of times: there was a matching thing that I thought was quite robust but really wasn't and there were things I needed to change. The software is based on a form of 'content-based music information retrieval' that I call phrase matching – that is, matching the sound quality of what it hears in performance to a corpus/database of material sampled from either the current performance, or previous performances. The matching process is the underlying algorithm that provides sonic coherence, and the software manipulates 'matched' phrases into new sonic objects for the performer to perform with. That's when I start thinking more methodically about being a planning software engineer rather than someone who just tinkers. It's the mistakes that make me have to structure my work.

Q: Do you think that you have in your head a set of qualities that you are looking for when you yourself are performing with the generative system?

B: Sonically and musically I am after something that's really rich, that's quite complex. I really like complexity and I like multiple layers of sound and for it to be difficult to know at first exactly what the difference is between the instrument and the system. Also, melody and harmony aren't the driving principles of this, it's very much the sound quality. So, in the first instance it's that, but it's also the system is using sounds that's it's heard in the performance and previous performances but the connections should not be very transparent, it should be reasonably opaque to me as the performer as to what it's doing. If it's too clear for me, I'm bored and I'm not provoked to do something different. And if the audience work it out within an eight to ten minute performance and they work out where those connections are, then for me, it hasn't been really successful. I think opacity is an important thing but I am always teetering on that edge: if it's too opaque, then it seems like random noise, if it is too easy to understand the process, and it's like I've understood what's happening here and I'm not interested any more. I feel that with some works- as soon as I've worked it out ... I like to have a little bit of a tussle with whatever it is I'm listening to or in terms of art, trying to understand. In terms of the style of the music, the context in which I perform these systems is usually in contemporary music where complex interactions that are atonal is very normal. But I find that if the interaction between myself, or another performer, provokes a sense of play and engagement, it sustains the performance.

Q: Tell me more about how you saw reflection as an approach in your work through the PhD.

B: I think it has definitely been embedded in my practice. The way I used it was on a number of different levels. After reading a bit about reflective practice later on I realised that 'reflection-in-action' was very much what I was doing. It is very much what practitioners do, when you are doing something you are assessing it and actually reflecting on whether it is going the right way. In terms of actually reflecting as part of my practice, I write a lot. At first it was note books and then started using Evernote as a brain dump space. It could have been that I was in a rut, I wasn't exactly sure where my work was going, I'd done a number of performances, I had some recordings, I might have listened to a few recordings and then started writing down a few ideas. Writing ideas about what I had been doing, where it is I think I wanted to go and how I was going to get there. Or it might be, 'there's something really interesting about this process of interacting with the software but I can put my finger on it' and then I would tease out that through writing. Some of those extended reflections became part of my thesis. I initially thought that this process of writing as reflective memos would be something that I would go back on and they would be very small, a couple of paragraphs 'on this day this is what I was thinking' but it wasn't very formal like that. When I would reflect it was usually because I needed a way to communicate to myself where it was my head was at. They were usually quite long extended reflections.

The way I've used the word reflection is that I use the practice or the problems of practice to jump off and starting thinking about broader issues, or to drill down into an issue that I've found out in my practice is a problem. Evaluation can be part of it I think. I might start off by evaluating my own work and then that will get me into a broader theoretical discussion about what it is to perform with a computer. For me that's broader than 'self-reflection' as I've termed it in my own work but the whole process is still part of 'reflective practice'. Coming from the seed of my own practice and evaluating what's been done I need to evaluate a bunch of ideas and yes 'evaluation' is a part of it- assessing whether or not something's working is definitely a part of it...But I see reflection as being broader.

I think that makes a massive difference to how you view a surprise, how you view an unexpected event. The unexpected event for the professional is a problem to be solved, whereas for the creative person, not everyone, some may focus on the craft skills rather than the originality side of things- most of those who are interested in creating something new- they see that unexpected event, as a challenge, an opportunity.

I find the process of creating software, creating composition, whatever it is, almost like you are trying to...this word comes up a lot you are 'problematizing' your own practice. Getting into the flow of creating something and multiple problems come up and they are interesting because you find that you realise that you are solving things you didn't realise you needed to solve and that actually advances your practice. I find the whole idea of creative practice as posing your own problems as being a big part of the practice.

My PhD was about self-reflective practice. I am also interested in finding things about people like myself are players and developing systems they interact with. A lot of musicians are creating their own software and interacting and I 'd like to do some research about what makes them tick and if there are any commonalities between the way people work that way. I think finding out about how people reflect is part of that as well.

Generative systems can be really complex and interesting on their own. As soon as you put a human in the loop, it completely vitalizes that whole situation and something that might not be that complex generatively or algorithmically, with a human in that situation, it can seem very complex and engaging from moment to moment. My system really- there's some complex things going on under the hood but it's not an artificial intelligence system at all, it's not working with A-life concepts, whereas there are a lot of system that are. But I think because it's setup to react and also to provoke certain things, that performance thing is very engaging...

Q: Are you saying you design the system to have a life of its own but not an autonomous life? It's designed for interaction with a human rather than designed to generate autonomous music?

B: That's right, exactly. I am fascinated by the field of computational creativity, but for me, it's never been about modelling, it's been about finding ways of provoking the human to act in ways that they might not with another human or they might not by themselves as well. Having that kind of feeding off each other is really important. It could be something really simple like: the system used to respond to an attack for instance, as soon as I started playing my saxophone it would start generating something but it would generate something in a very opaque way so I would perform, it would respond but the response was very unpredictable. After a while I realised that that process of interaction was predictable. Even what was happening when the system did start, was unpredictable, the relationship between me and it was always: 'I play, it responds etc'. After a while I started putting in some autonomous, self-referencing things so it could go off on its own, I could stop and it could generate things so in way that was autonomous but it needed seed material. It needed something from a performer to work and that's what allowed it to go on.

Q: Am I right in thinking you work individually with your software but perform collaboratively?

B: There have been a number of collaborators at different levels: a couple of sustained collaborations so one is a very good friend who is also a saxophone player, and a good improviser and he has performed with it a number of times. He is also really interesting in that every time he has performed with it he had had suggestions, things he's wanted to add to the software. It wasn't that I took on every one of the suggestions but there was one aspect of the software that I'd built into it that over multiple sessions, every rehearsal you could save data so you could get on the stage and the software would have sound materials to work from that had been built up over numerous interactions. That kind of collaboration was sustained. Then there are people who have performed with it by downloading it from the website.

Q: Are you conscious of a kind of embodiment when performing whereas when you are writing a piece of software using a keyboard, there's a degree of separation?

B: Definitely. When I play my saxophone, it is immediate. If I am improvising on my saxophone, it's very moment to moment. I have skills on the saxophone that I've built up over

many years so it's very much embodied, interaction with this 'thing'. In the same way I play a little bit of guitar, it's very rare these days but when I pick it up that feels very different to the saxophone but, at the same time you are exploring that instrument and playing with sound directly. Whereas with software, you have to put something down- it's very much you are creating something and then assessing it after... You are externalising, encoding an idea of what might be interesting but it's not embodied in the same way as playing a saxophone for sure. But some of the tools I develop I try to embed performative aspects to them so something that is a little bit generative or a little bit randomised that I can still play on a Midi keyboard or something like that. There is a feeling that I am actually affecting the computer software but it's much later on in the process.

-
- ¹ Carey, B. (2016a). '_derivations and the Performer-Developer: Co-Evolving Digital Artefacts and Human-Machine Performance Practices', PhD thesis, University of Technology, Sydney, Sydney.
- ² Carey, B. (2013). '_derivations: Improvisation for Tenor Saxophone and Interactive Performance System, *Proceedings of the 2013 ACM Conference of Creativity and Cognition*, Sydney, Australia.
- Carey, B. (2016b). Artefact Scripts and the Performer-Developer, *Leonardo*, vol. 49, no. 1, MIT Press: Cambridge MA.