don’t touch my MIDI cables: gender, technology and sound in live coding

Joanne Armitage and Helen Thornham

abstract

Live coding is an embodied, sensorial and live technological–human relationship that is recursively iterated through sonic and visual outputs based on what we argue are kinship relations between and through bodies and technology. At the same time, and in a familiar moment of déjà vu for feminist scholars, live coding is most often discussed not in relation to the lived and sensory human–technology kinship, but in terms of fetishised code or software, output and agency. As feminist scholars have long argued, emphasising and fetishising code or software, and celebrating output and agency are normatively masculine, white and Western conceptions of technology that feed into the growing valorisation of accelerationist logic whilst also negating embodied, not to mention other (non-white, Western, masculine) bodies, expertise or histories per se. In this article, we want to redress this by drawing on our empirical material on live coding to focus on human–technology kinship and, in so doing, think about failure, slowness and embodiment and about human–technology relations that are more akin to what Alison Kafer (drawing on the work of Donna Haraway) has termed ‘becoming with’ or ‘making kin’. This, we argue, has the potential to shift the focus from the potentialities of technologies on or through the body, towards the generative capacities of mediation (including failure), which are caught up in lived experiences. The question is not only about how the relations of bodies and technologies are played out in certain circumstances but about what might be played out if we reconceptualise these relations in these terms.

keywords

live coding; gender; sound; technology; code; cyberfeminisms
Live coding is an embodied, sensorial and live technological–human relationship that is recursively iterated through the sonic and visual in ways that direct us to what Alison Kafer (2019, p. 6) has termed (drawing on wider work on kinship, e.g., Haraway, 2008, 2016; Rodriguez, 2014; Benjamin, 2018; Murphy, 2018) ‘becoming with’ or ‘making kin’: a lived relationship that at the very least challenges ‘the idea of wholeness’ through a focus on ‘coalitions’ and ‘interdependencies’ (Kafer, 2019, pp. 5, 6). It is a conceptual, ideological and political intervention into patriarchal and colonial structures of (re)production. Alison Kafer’s work on kin sits within disability studies and focuses on human–technology relations, but the wider concept of making kin is threaded through (for example) queer theory (Rodríguez, 2014), race and ethnicity studies (Burch, 2014; Benjamin, 2018) and cyberfeminism (Haraway, 2016; Clarke and Haraway, 2018). What connects this work is a concern with power relations and rethinking mediation and lived relations in ways that are attuned to bodies, senses and relations. Indeed, as Adele Clarke and Donna Haraway (2018, pp. 32–33) argue, making kin is not only about ‘creating a new relationship’ but also ‘refers to daily actions that transform partial relations into deeper ones’ over time and through the everyday.

At the same time, and in a familiar moment of déjà vu for feminist scholars, live coding is most often discussed as a technologically mediated production that has a creative author or agent in control of output. It is authorial or creative output and agency that is celebrated, whilst code and software are fetishised. These accounts are problematic for us as feminist scholars, not least because they bolster normatively masculine, white and Western conceptions of technology that in turn feed into the growing valorisation of accelerationist logic, whilst negating embodied experience, not to mention many other (non-white, Western, masculine) bodies, expertise and histories (see Grosz, 2001; Wajcman, 2004; Noble, 2017; Benjamin, 2019; Bassett, Kember and O’Riordan, 2019). This in turn understands technology in oversimplified ways as a tool or facilitator, or problematically as a benign and apolitical support for the agential human. In this article, we want to redress this imbalance by drawing on our empirical material on live coding to argue that live coding is better thought about in relation to making kin. By comparison with existing studies of live coding (discussed briefly below), we want to suggest that live coding disrupts traditional and long-running conceptual separations of the body and technology, directing us to think about failure, slowness and embodiment and about human–technology relations that refute notions of whole agential subjects.

live coding, a (problematic) introduction

Live coding is part of the wider electronic music scene (see Rodgers, 2010; Amrani and Payne–Frank, 2017; Armitage, 2018), and comprises improvised sonic, visual and multimedia experiences using code. At live coding events, code is projected onto a screen behind the live coder, who ‘writes’ on their laptop through a range of software. The code ‘generates’ sounds and beats, music and rhythm, and the live coder ‘builds’ music in front of a live audience. Live coding languages are normally open source and have been developed by live coders for their own practice. Some live coding environments are used widely, such as TidalCycles, SuperCollider and FoxDot, whilst others are more idiosyncratic, such as Renick Bell’s Conductive.¹ Sounds emerge from repeating and editing code patterns that move, iterating

¹The labour and time required to create, package, maintain and document their creations means that wider distribution is challenging for many developers. On his webpage for Conductive, Bell speaks directly to these issues, highlighting the importance of institutional support (http://www.renickbell.net/doku.php?id=conductive [last accessed 15 December 2019]).
cyclically, changing or shifting every X number of times to form sequences of notes with variations. Another approach is where oscillators are patched together, nesting and merging to form new sounds. These practices resonate with other types of electronic music 'performance' where sounds can be sequenced (i.e. a drum machine) or modulated (i.e. a modular synthesiser). During some live coding events, code is written from scratch as an intrinsic part of the experience, and language like TidalCycles and ixiLang have been built with this in mind, employing a succinct syntax that can be typed quickly. At the other end of the preparatory spectrum, we find 'clean' and prepared code that is triggered and tweaked through the set. At a space in between, little bits of code are used and worked with to form something new by layering it up, by iterating, deleting and moderating. This means that the 'practice' of live coding differs (see Armitage, 2018), with some live coders generating sounds using mixtures of samples, audio hardware or software synthesisers.

To date, the dominant conceptions that emerge within scholarship on live coding continue to emphasise authorial/creative agency over technology, or authorial/creative agency over sound and time/space. Earlier research on live coding, for example, is threaded through with accounts of a 'performance' by a whole agential subject who creates and builds sounds and technology to their own vision. If we think of work that considers live coding as a musical and creative practice (e.g., Magnusson, 2011, p. 21), for example, live coding is understood on the one hand as the 'offspring' of formal music composition that has always been technologised (e.g., scores), and as the live 'performance' of those technologies (complete with capacities for disruption and nuance) on the other. This introduces notions of play, disruption and generative capacities through live 'performance', but such generative capacities are realised through an affective response to the 'performance', and the live coder remains, as Thor Magnusson (ibid., p. 22) argues, 'primarily a composer'. Notions of agency and authorship are prioritised here, even if the 'performance' itself is nuanced to incorporate doing code in time with sound and rhythm.

Work that understands live coding in relation to the development and use of tools and technological systems with which to create and perform (e.g., Collins et al., 2003; Roberts and Kuchera-Morin, 2012) also sets up technology in keeping with normative conceptions of it as a ‘tool’ to be manipulated, and built. Charles Roberts and JoAnn Kuchera-Morin’s (2012, pp. 68, 64) article, for example, details the functionality of Gibber, a JavaScript library built for ‘authors’ and ‘performances’, and the conception throughout the paper is that the JavaScript library is a benign, apolitical 'tool' that can be shaped to the coders’ desire. Here, agency and mobility reside with the coder, who chooses the tools and the processes of working with them. The software is understood as a vehicle, arrived at through a weighing up of logical and rational decisions by the user around functionality: the software is a tool or terrain to be picked up (or not), and moved through (or not), but the decisions and agency reside with the human.

Accounts of live coding that understand it as a 'performance' involving improvisation, notation and composition (Parkinson and Bell, 2015; Cardenas, 2018) produce similar discourses, with the technology

---

2 Heavy Lifting (https://heavy-lifting.org/ [last accessed 15 December 2019]) creates her own samples, and sequences patterns of them using TidalCycles. Artists such as Marianne Teixido (see https://marianneteixido.github.io/ [last accessed 15 December 2019]) perform by merging different waveforms in SuperCollider.

3 There are notable exceptions of course, such as the work of McLean and Dean (2018) who argue for a rethinking of code, Tara Rodgers’ (2010) interviews with artists within the electronic music scene and Emma Cocker’s (2016) rethinking of temporality and liveliness in live coding.
understood as either a 'musical tool' in and of itself or a 'reproduction' tool, whereby pre-written and imagined scenarios (in the form of code) are played out to a 'script' (Parkinson and Bell, 2015, pp. 1, 8). Such accounts place live coding on a continuum with other live music, emphasising the notion of performing or performance whereby the live event becomes the playing out of intentionality in an ordered and coherent way. This enables live coding to be compared with other live music events and, in so doing, it is the comparative elements—for example, performance, output and composition—that are emphasised.

In the 2019 documentary on live coding Generation Algorave (2019), published on Resident Advisor, it is cleaner code, smarter and faster ‘performances’ and clearly authored content that are celebrated. The documentary tells us that artists 'harness' algorithms to 'make live and improvised electronic music'; they 'broadcast', 'run processes' and 'create sound in real time'. All of these claims celebrate the agency/creativity of the live coder and conceive the technology as manipulable: a tool to be constructed and not (for example) embedded with discriminatory logic (Noble, 2017, p. 1) or producing inequalities (Thornham, 2019).

Such accounts of live coding understand it as a technologically mediated production that has a creative author or agent in control of output. More than this, the technology is understood as a facilitator or support to the realisation of an author’s/artist’s vision. Understanding live coding as a ‘performance’ feeds into this, conceiving the event primarily as the execution of intention, or a negotiation between a human agent’s intention and execution. All of these accounts are highly problematic not only because of the location of power and control with the human, but also because of what this means for how we are then able to think through technology, mediation and affect. They are also problematic because they are far removed from the experiences recounted by the live coders we interviewed, and indeed far removed from our own experiences as live coders within the live coding community. Instead, and as Tara Rodgers (2010, p. 249) argues, these accounts offer a ‘double standard’ whereby ‘slick production’ is what is valued even as innovation or ‘do-it-yourself’ might be overtly celebrated. ‘Slick production’ feeds into the wider fetishisation of technology (ibid.) and, as scholars such as Jodi Dean (2008) remind us, it rarely interrogates power relations and leaves unexamined the valorisation of the white and masculinist qualities of speed, smartness, smoothness and autonomy that are (also) heavily enmeshed with the values of neoliberalism (Braidotti, 2013; Wajcman and Dodd, 2017). That the body is set aside in all these conceptions of live coding might also be of little surprise to feminist scholars: this is another déjà vu moment where technology and agency are the valorised qualities to the detriment of (and separated from) bodies (see Grosz, 2001; Thornham, 2007). What is being valued in these accounts of live coding, (and by implication) for whom and by whom, are therefore important political questions (see also Noble, 2017; Benjamin, 2019).

In what follows, we have structured the article to consider a number of issues that have fed the more normative accounts of live coding discussed above: the idea of slick production, or the idea that live coding is the realisation of authorial intention, is discussed immediately below. Following this, we focus on how live coders conceive of themselves and their work in order to suggest that the idea of a ‘performance’ also needs rethinking. Having looked at live coding bodies, we then think about technology and code in order to question the idea of it as an apolitical facilitator for creative agency or fetishised technology. Finally, we think about the sonic elements of live coding, asking how we should rethink this
away from notions of output. Each section draws on empirical work with live coders and is designed to question and problematise the accounts of live coding discussed above.

**slick production, reconsidered**

We start slightly unconventionally with a diary excerpt of a live coder during a live coding event in Osaka, Japan in 2018. The extract is one of a number of diary entries that were made during a project that included a series of live coding events and workshops in Japan and the UK. In addition to the diary extract, we also draw on interviews conducted in 2017 with eight women (then) active within the UK live coding scene and performing at Algoraves. All artists were based and working in Europe. Of those interviewed, five artists were British, one Belgian, one Iranian and one Latin American. The Algorave scene in Europe has origins in academia and brought together highly skilled computer scientists who were majority white men (see Armitage, 2018). Although originally relatively homogenous, the Algorave scene has opened up to the possibilities of live coding as a practice and a community. Our interviews were conducted during a time when women performing in the scene had become increasingly visible (see Bolt, 2017). Whilst visible in one sense, our interviews revealed ways in which legitimacy was always questioned and questionable. Women Algorave artists have led numerous live-coding workshops for women and non-binary participants; three of the interviewees had attended these workshops before performing at shows and went on to lead workshops themselves. Other women engaged with the scene through academic and arts/technology networks. Those interviewed were all university educated, some with postgraduate degrees. Most had no formal training in computer science; several were in the process of or had completed postgraduate degrees in music. The whiteness of the UK (wider European and North American) scene is notable in the context of a collective that positions itself as global, inclusive and diverse. Latin America is a site of much creative and technical innovation in Algorave, drawn together by collectives such as Colectivo de Live Coders (CLiC). Still much of the capital in and power of Algorave is centred around white European academic institutions. Exchanges between different global ‘nodes’ occur through online performances and various conferences and events. The diary extracts, as detailed, were part of an exchange between artists in Yorkshire and Tokyo.

Interview questions were semi-structured and focused on artists’ creative and technological practices as well as their experiences of performing within the Algorave scene. Discussions took place using a real-time text-editing platform (PiratePad, now defunct). PiratePad enabled interviewees to review, extend and make alterations to their responses after the meeting. The excerpt below sets the scene, detailing the complexity of human–technology relations, as well as a range of power relations, imaginings and feedback that complicate the dominant notions of live coding discussed above: as some sort of technologically enhanced ‘performance’ where an artist—through their enhanced digital expertise and artistic vision—transparently executes ideas to produce a material output for an audience to interpret:

This project was led by Alex McLean and Renick Bell, supported by the Daiwa Foundation, the Sasakawa Foundation, Arts Council England and the British Council.

The diarist was not an interviewee.

CLiC, https://colectivo-de-livecoders.gitlab.io/ [last accessed 10 August 2020].
Opening the case, she begins untangling wires: those to power, those that communicate between and route sound from devices through the laptop and to the PA. Talking to herself and speaking it out ... Locating a power supply to get sound produced as quickly as possible through the PA and headphones. Two synthesisers go into the audio interface, then USB cables—grey to the MIDI hub, and black to the audio interface. The MIDI hub has powered up and the laptop is recognising the audio interface. Her MIDI cables are connected from the hub to the synthesisers. MIDI Out to MIDI In connecting to channels one and three so it matches with the code. She presses her drum machine *boof* *boof* levels flashing as sounds move through the system and to her headphones.

Her screen is hooked up to the projector through HDMI—a flicker and her screen is on display. Font colours changed to black and she resizes it. Her server boots as the text at the bottom of the screen flashes green. Initialises MIDI in the code and sets the tempo, quantisation, crossfades. All seems well. She runs the code for sound playback but there is no sound yet. In a new window she writes a pattern to play a sound. Listening through headphones. Nothing. Looking for light, flashing at MIDI box—signal received there. No MIDI received on synthesiser, no red flashing light in time. She checks connections, rewires hardware. Runs again. She tests the drum machine, a few taps on the machine itself ... The first pattern is copied and pasted and routed to the drum machine. No sound. She looks up briefly at the room, unmuting the external mix. No sound. Crashing.

A man comes across, touching the cables. She sighs and looks frustrated, takes her cables back and continues to work watching for flashing lights and signs of sound. The crowd—made up of mostly other performers like me and those who have attended a workshop earlier in the day—seem supportive but uncomfortable, twitchy. Another performer comes to the edge of the stage about to play, ending the awkward silence. She reconfigures her audio routing through some proprietary software and the other performer leaves the stage. She plays.

(Live coder's diary extract, 2018)

The extract above is a rich first-person account of a live coding event that we are citing at length because of the level of detail in the technological set-up and the focus on process rather than (just) output. There are a number of interesting elements in the above, including the noting of procedural technological-embodied ‘steps’, which are rapidly revealed as responsive rather than pre-emptive as the live coder attunes herself to the systems, taking care of wires, cables, plugs, (not just) code, syntax, grammar. It is an empathetic account, detailing hesitations, iterations and responses from a knowledgeable observer position, and we have cited it at length here as a way of introducing a number of issues that strongly resonate with wider issues within feminist scholarship that we will touch on in this article: technological failure by comparison with agency and authorship that asks us to rethink power relations and dynamics as well as conceptualisations of technology, speed and bodies. Related to this is the issue of where failure happens—at the site of the embodied—and what this does (and has done) to conceptions of technology, which through this remains problematically fetishised. Clarke’s (2018, p. 33) notion of kinship as relations ‘crafted ... over time’ seems pertinent here not least because the diary extract details live coding more in keeping with mundane and sensory logics, or embodied relations. All of this, of course, is part of our attempt to rethink live coding not as output or performance, but as lived and relational, embodied process.

**performance, reconsidered**

As suggested above, conceiving the live coder as powerful and agential maps onto wider and long running problematic constructions of the (normatively male, normatively white) user of technology as a whole liberal subject, able to navigate through technologies, contribute and shape content and
choose.\footnote{For a discussion of these issues, see Gómez Cruz and Thornham, 2015.} As Helen Thornham (2019, p. 5) has argued, this ‘free’ and ‘autonomous’ neoliberal subject is ‘the fantasy of both gendered subjectivity and technology’. More recently, scholars such as Ruha Benjamin (2019) and Safiya Umoja Noble (2017) have drawn on empirical research to extend these discussions beyond gender to issues of race and ethnicity, not least because they align with and reproduce neoliberal, white and modernist ideologies (see also Ricaurte, 2009; Couldry and Mejias, 2019) that continually ‘see’ and validate certain interpretations of users and mediations (as with the accounts of live coding above, authorship, agency and control), whilst negating others (such as embodiment, sensory experience and mediation). In addition, George E. Lewis (2018, p. 128) argues that accounts of computer improvisation that focus on the agency of the ‘performer’ are based on very shallow interpretations of technology, coding and even ‘performance’, which fail to understand improvisation (amongst other things). For Lewis (ibid.), computer improvisation shifts creative practice away from the assumption of an innate quality of the artist who has agency and insight, towards an understanding of a scenario where ‘alternative points of view’ that are both machinic and human are ‘encountered’.

For our interviewees in 2017 as well, the experience was as much about response and reaction to the environment, which included sound, the crowd, technological nudges and demands, as about silences (for example). Interviewee C told us, ‘I want to respond live and try to adapt’, not least because her live coding ‘is completely improvised’. Her understanding of live coding was bound up in ideas of responsiveness and reaction—of adaptation—and not in notions of ‘performance’ or preparedness. This was a similar refrain across our interviewees; Interviewee B refused the label of ‘performer’, telling us: ‘I, by default, am NOT a performer’. For her, such a concept was bound up in notions of process and output, and as she told us: ‘I’m not a performer, I’m not making things that are meant to be functional. That’s the thing, and I love that. It can be useless, but the things are tailored to that moment and that specific imagined demand’ (Interviewee B, September 2017).

The account above, at the very least, asks us to think about what we are valuing when we label a live coding event a ‘performance’, and while the term has different connotations depending on context, the idea of a performance necessarily constructs the event in ways we (and some of the live coders) find problematic. For the live performer cited above, the ‘thing’ that is created comes from nudges and demands that are technological and environmental, as well as imaginings of audiences and of herself, and out of a temporally specific moment that the ‘thing’ is trying to speak to. This works against notions of a ‘performance’ then, not least because it positions the live coder as interdependent on and within these various elements, as well as responsive and reactive to them. ‘Improvisation’—within these terms—is also problematic as it still maintains an element of human agency and direction. In the account above, she refutes the idea of herself as agential. N. Katherine Hayles (2017, p. 109) reminds us that the perception of agential control also problematically posits technology as responsive to and initiated by the user and as subsequently benign, apolitical, open, causal and linear, when in fact we need to be thinking not about control over or through, but about ‘the ways in which assemblages come together, create connections between humans and technical actors, initiate, modify, and transform’. The interviews with live coders also articulate live coding experiences as iterative, attentive and responsive (never as anything resembling agential or authorial/creative power). Interviewee B told us that all of her
live coding experiences could be described as her being on the precipice of crashing, and it was this experience that not only shaped her live coding but that she had also come to enjoy and celebrate: ‘it’s a weird type of confidence, to not care when things go wrong, but I like it. It feels honest’. In this account, crashing is described as a lived context that makes a series of demands that require a series of responses (always unintentional) and that are embodied and lived (‘I enjoy feeling unstable’, she tells us). Thinking of about-to-crash as a positionality rather than an outcome or counter-outcome (death/end of the ‘performance’) feels ‘honest’ for this live coder, not only because it better reflects the responsive, embodied, improvisory experiences but also because, for her, it refutes the claims to power, authorship, agency over the technology.

The experiences of live coding noted through the diary excerpt above also discuss rhythms, tones and beats that generate a range of responses as the live coder plays or alters them in a continual and active negotiation and creation of and with the technology, the code and the sound. The tracks and traces, loops and iterations are overwritten, deleted, expanded and sedimented in terms of embodied processes and practices that repeat gestures and touches, methods and loopings. The live coder is positioned within power relations of the technology, including the sonic outputs of that technology that interpellate her and demand action, response and attention from her. In the diary excerpt, it is the attention to/through the headphones that demands action; the flashing lights demand different responses. The lack of sound produces further checks and edits in a continual iteration. This is a ‘profound politics of small things’, to quote Shannon Mattern (2018, p. 84), not least because technological materiality, spatial and technical architectures and temporalities all form the ‘conditions of possibility’ (Suchman, 2007) in which the live coder is positioned and must work. Live coding here is not an output or a performance, but needs to be thought about more in relation to, following Kafer (2019), interdependencies that are differently and variously felt: sonic (e.g., sounds, lack of sounds, glitches, tone, speed), linguistic (e.g., code, discourse, audience), visual (e.g., flashes, crashes, screenic) and embodied (the crowd, the architecture of the room). It is these networked interdependencies that co-create the conditions of possibility for the live coding experience as always-already human-machine: the experiences recounted here detail deep affective ties (Rodríguez, 2014, p. 49), and are far removed from the accounts of live coding discussed above as performance or creative output (for example).

**technology, reconsidered**

The section above discussed the experiences of live coders in ways that problematise thinking about live coding as a performance. Here we want to turn our attention to technology. Live coding produces and works with glitchy systems, in part because of the liveness of the event but also because of the open source resources and the continual hacking and hackability of these. These glitchy and failing systems also enable us to unpack and create tensions with(in) the powerful narratives of technology as facilitator for creative agency or as fetishised code or equipment noted above. TidalCycles, for example, has been hacked and rewritten by and for live coders in particular ways for collaboration (Estuary⁸), less syntax (Siren⁹) and interoperability (AIOI¹⁰). These extensions are often built to address a particular

---

⁸Estuary, https://www.d0kt0r0.net/projects/index.html [last accessed 20 February 2020].
⁹Siren, https://github.com/cannc4/Siren [last accessed 15 December 2019].
¹⁰AIOI, https://github.com/MAKIO135/aioi [last accessed 15 December 2019].
challenge—live coding languages are limited and restrictive. These processes remind us of the continual iterative practices that entwine software, code, embodiment and experience. Hacking code and software means that live coding has the potential to be more esoteric and collaborative, offering specific avenues of intense subcultural deviation. It also means that we need to think of code, software and live coding as always in process, always being generated and produced, never fixed. If we think of code, software and live coding in this way, we move away from understanding live coding as a single event or ‘performance’ and shift towards a notion more akin to Kafer’s (2019, p. 6) ‘networked interdependencies’, where temporal and spatial elements also need considering (live coding as a moment, a node within a wider network, or an assemblage within assemblages). Many of the live coders understood code, glitches and crashing in embodied ways: not so much as an issue with the code (over there), but more akin to an embodied moment of latency.

I have plenty of code errors that I have not been able to react to in the heat of the moment and afterwards have kicked myself about. I have some saved bits of code I can quickly use if I freeze or something is not working. I had an audience member mansplain brackets to me after I froze and missed one and could not see it for ages. I just nodded and said ‘thank you’. I was not using Atom at this stage but was straight into Cyril. (Interviewee C, September 2017)

In the account above, what is interesting in the context of this article is the way that Interviewee C conceives latency as an embodied experience (‘I freeze’, ‘I froze’), and there are resonances here with, for example, Diane Carr’s (2002) work in Games Studies, where she argues that failure is tied in highly gendered ways to bodies for women gamers, whereas failure for male gaming bodies is tied to narrative choice or technology (these arguments have also re-emerged in relation to fitness apps and quantified bodies; see Thornham, 2019). The account of live coding in the excerpt above is complex, incorporating software (Atom, Cyril), discursive lines of code (code error, ‘saved bits’) as well as (and alongside) the (gendered) sociocultural (an explanation of something going wrong where ‘mansplaining’ is as relevant as code error, saying ‘thank you’), and embodiment (freezing). Hayles (2017, p. 121) argues that systems are ‘hardwired to prevent latency’, but as the extract above and other live coders remind us, latency is not only a familiar experience in live coding (‘I crashed. Cue empty tent’, Interviewee D; ‘I once stood in silence for 15 minutes’, Interviewee H; ‘I’m just responding to code errors’, Interviewee C); it is also an embodied experience, and one that produces gendered and socially ‘normative’ responses. In the excerpt above, ‘an audience member mansplains brackets’. In the first diary excerpt, a man comes over and picks up and touches her cables. In another account, Interviewee E is asked to prove her technical competency: ‘like was it really me doing the coding? Prove it’. We could read this in a number of ways. We could think, along the lines of scholars such as Elizabeth Grosz (2001), that latency is experienced at the site of the body in a familiar moment of déjà vu in which the body is measured against the speed of the technology (slow body, fast code) and found lacking (‘I was going to say “somebody”, but it is always men—undermining me or questioning me’, Interviewee F). We could read it more positively in terms of a

---

11 There are caveats here of course in terms of literacy as well as the conditions of possibility that might shape what and how an event might develop or emerge.

12 We are not celebrating open source here, which is open to many of the same criticisms around white, Western, masculine visions of technology. However, open source software and code does mean that the systems have more capacity for mutation, impeachability and failure because both the code and the software can be hacked (with the caveat that this might still be to a certain white, Western, masculine vision of authorial agency or creative intention).
necessary resistance to accelerationist logics of technology that insists on understanding technology as embodied, a counter to fetishistic notions of technology (like in the accounts of women in the electronic music scene interviewed by Rodgers, 2010). We could think about the way this account extends our understandings of code always already beyond the discursive to (and with) the embodied. All of these are important to consider, but what is most relevant for us here is that this is an understanding of live coding that embeds bodies, code and technology together, and that (again) conceives the live coder as attuned and responsive, not as directive.

If live coding is about attentiveness and being attuned, it might be more fruitful to position it within different histories of technologies— as part of a much longer trajectory of skilled technologists who repair machines (McNeil, 1987; Cockburn, 1992; Jarrett, 2017; Hicks, 2018). This might be more appropriate not least because it circumvents and intervenes in the histories and ideologies underpinning accelerationist and productive narratives of technology (see Bassett, Kember and O'Riordan, 2019), but also because it asks us to shift our understanding of what is going on from some kind of ‘performance’ to one more akin to ‘care’ or attentiveness in a similar vein to the live coders who also talk about live coding as reaction and response (as above). Seen in this light, the live coder is constantly attuned to the systems and sounds, working at a range of temporalities and materialities. These temporalities do not just relate to technological speed in a live coding scenario: they also relate to the ‘living present’ (White, 1987) of a practice over time as well as the sonic and material elements to which the live coder is also responding and which are bound within and beyond the technical. The recreations and reimaginings that occur as a consequence of (different) human–technological responses redirect and nuance the experience and the sounds ‘containing’, as Steven J. Jackson (2017, p. 179) argues, the marks of breakdown or failures of the systems traced through red text denoting errors on the screen and the reordering of cables. In this way, traces and archives are collated—but as process or recursive loopings—not as searchable/writable code or software.

code, reconsidered

Thinking of live coding in these ways also makes us rethink code and coding. Code is written, edited, cut, copied, pasted and re-edited through movements that rapidly go up, down, backwards and forwards through the lines and pages of code. The systems themselves are either bespoke—made for live coding (i.e. TidalCycles, FoxDot and Gibber)—or generic programming environments that have been adapted with libraries so that they can be used for live coding. Here code takes multifarious forms: fast, terse and minimal; representative, graphical and symbolic; verbose, clear and human-readable. The terseness of code is often an important factor in the design of live coding languages due to the pressures of live coding, speed of typing and the legibility of syntax—in code with many nested brackets, it can be hard to spot an error (as the account above details). Here speed is prioritised, but the shrinking of syntax—fewer brackets, commas and semicolons and shorter function and variable names—can make code less legible. There are many variables here—none of which posit a notion of code as ‘smooth’ (Rodgers, 2010, p. 249). But when we add to this the environment of live coding, code becomes illegible, incoherent, non-linear, paused, recursive. In a live coding scenario, code is prepared alongside the executed code in a multiplicity of temporalities and speeds: this is a myriad of operations, of ‘strange temporalities’ (to borrow from Halberstam, 2005, p. 1), of embodied and generative actions: more like
continual and iterative recursive looping (Day and Lury, 2016, p. 43) than a linear story. In keeping with critical software studies that argues for the dynamic elements of code (Cox and McLean, 2013; Cocker, 2018), live coding does not uphold a notion of code as a linear, taken-for-granted, separate or distinct language that is 'over there', not least because a significant amount of any live coding experience is taken up with iterating code, linking or patching code, locating breaks within code and looping code.

Indeed, Emma Cocker (2018, p. 8) argues that code is 'folded' into the output, and her notion of 'kairotic coding' is an attempt to embrace the openness, live(li)ness and potentialities of coding. For Cocker, live[li]ness refers to notions of 'living present' (White, 1987) and 'thinking-in-action' (Cocker, 2016) to move away from a focus on the semantic feedback of code and notions of code as predictive (arguments that see live[li]ness as the product of fast[er] and more dynamic systems) towards a notion of code as unexpected and unpredictable (see Cocker, 2018, p. 83). Also according to Cocker, live coding involves writing or editing code at a variety of speeds, working at different temporalities—planning and editing before, and listening after the execution of code, so sound is building, evolving and fading. Such arguments prioritise unpredictability and uncertainty as well as multiple temporalities in which the live coder is located. This suggests that we need to think of code—like technology and systems—as impeachable and questionable rather than as 'building blocks' for technology with connotations of solidity and fixidity. For the live coders too, code is much much more than a language or communicative ‘tool’; it is more akin to an organising principle that extends to many aspects of life:

I feel that if I don’t organise myself precisely with expected variables in my own life (x amount of sleep, for example), then things can get chaotic. But managing these variables can make things a little more expected and easy to maintain. Splitting things up into functions. It mirrors the way I think a lot. (Interviewee B, September 2017)

Here, the organising principles of code are felt much more widely as the interviewee discursively maps the logics of code (bound in her notion of the 'variables' and 'functions' and how they are put into practice) across other aspects of her everyday. This resonates with a number of interviewees who talked about code organising, or code as a particular viewpoint or method, and that live coding was a ‘natural’ activity that extended what they were already doing or thinking. Interviewee A talked about live coding as ‘meditation’, a daily ritual. Interviewee B told us, for example, that she does not know which came first, 'this "computer logic" way of thinking or the coding as a hobby itself'; she explained that her preparation for live coding was unnecessary because it was ‘an extension to my life. I dream in neon cubes and lambda’. Interviewee G talked about how she reimagines and merges her interests in cosplay and Japanese IDM through her audiovisual coding practice. All of these conceptions of code are far removed from notions of code as a ‘tool’, or even a ‘language’, and extend it well beyond the linguistic and discursive into the ideological, if not the imaginary.

At the same time, it is worth noting that the event of live coding continues to fetishise and value code, abstracting it (despite these experiences of live coding) from the body (of the live coder). In thinking about the set-up of live coding events, it is worth noting that it is the code that is showcased, projected onto the screen, yet again (as Rodgers, 2010 reminds us) represented as vital and as having vitality, fetishised, abstracted and represented as ‘smooth’. As suggested at the start of this article, this is a
'double standard' that visually separates, celebrates and promotes technology-as-code, representing the code—rather than the live coder—as fast (even if code operates at strange temporalities, it is still fast). On a large screen often projected over the (smaller) body of the live performer, the code seemingly writes itself (doubly emphasised by the invisible cursor on a black screen with red or white font), while the body of the live coder is represented as a surface, projected over and onto, disappeared, negated: not even really the tool for the code, but more like an excessive accompaniment (see Figure 1). All of this creates a sense of value: feeding already widely normative notions of technology as fast and powerful whilst the body is left behind: negated, unimportant. Visually then, this set-up is all about the technology and about output (further contributing to the persuasive and normative interpretations discussed earlier in the article). There is something deeply at odds then with the representations of live coding that value code, speed, technology; and the experiences of it as lived, kairotic, interdependent and embodied.

**output, reconsidered**

In the penultimate section of this article, we want to ask how we should think of the sonic elements of live coding, suggesting that the sonic (like code and systems and the experience) should be thought of in relation to concepts of kin. Hopefully it is already clear from this article that our conceptualisations of the sonic elements include the technical and embodied: sound is a technology and an output, a
contributor and a consequence. It is messy, and focusing on the sonic showcases much more of an ambivalent relationship with technology not least because unwanted tones and sonic accidents make disruptive and immediate demands on the live coder and rupture the ‘smoothness’ associated with notions of ‘performance’. For us, the sonic generates and co-constitutes these experiences and is interdependent with processes of interpellation, discipline and biopower discussed at the start of the article. At the same time, thinking through the sonic elements of live coding highlights a number of issues around temporality: how we might better think of sound, and the relationship between sound and expertise that is, of course, highly gendered.

One way we need to think about the sonic is in relation to temporality. Temporality is a great concern to the live coder who experiences time in many different ways and, as already suggested, the work that live coders do is prehensive (to borrow Lois McNay’s [2000] term), ‘anticipatory’ and ‘predictive’ (to borrow from discourses on big data; see Andrejevic, 2011; Berry, 2014): performers are just ahead in terms of preparing code for the about-to-be sound, structuring and restructuring sound in pretensive not just retrospective ways. Performers are also, of course, in time, in terms of ‘playing out’ material forces that are regulated by time (see Grosz, 2005, p. 181). Seen here, live coding is an enmeshing of temporalities that ‘explicitly foregrounds the real-time character of computational processes’ alongside and through real-time ‘thinking-in-action’ of the live coder (Cocker, 2018, p. 84). The execution of code is crucial here; the moment of change, the termination of code are moments that pivot between the temporally specific act of coding and the material playing-out of sound in time. This is not linear even if it is processual. Existing concepts that engage with theories of time through accelerationist discourses of technology seem woefully lacking here: this is not the timeless time of Manuel Castells (2010 [1996]), the instantaneous time of John Urry (2000) or the network time of Mark B.N. Hansen (2009), for example. Nor is this the constant pressure towards productive time that Judy Wajcman (2019, p. 317) discusses. The live coder is constantly listening—is attentive and attuned to (not just the machines and systems as above but also to) the sounds—but the demand of thinking through code distorts this as a reflective practice. At the same time, the cycles of live coding create change and interest in sound—generating polyrhythms by overlapping values for musical material (rhythms and pitches) so they move in and out of phase, a three over two or five over seven. In this context, live coding is making, editing and responding to soundwaves: the changes in sonic energy over time. Synthesised waveforms that also break and flow differently; the register and texture of layered sounds that can rupture and clash; rhythms that can stutter in and out of time, repeating unpredictably; qualities of sound manipulated through whirling numbers, unruly, unpredictable and unimaginable. The relationship between code and waves is intangible, but both are constantly moving and collapsing into each other. Change over time, avoiding moving too quickly or slowly over time, keeping the waveform dynamic, are all concerns of the live coder, who reacts to beats, rhythms and sounds as much as technological nudges, or the ‘feeling’ of the crowd. The excerpt below details a range of cognitive and ontological responses that in turn highlight multiple conceptions of sound—as affective (‘break the groove’), as rhythm (‘bop bap boop’), as layered (‘timbres’), as process (‘flow’), as imagined (‘I try to think’), as disruptive (‘too loud’) and so on:

Sounds come in and out. Some break the groove or come in too loud, and I’d rather they come in and gel a bit with what’s already there. I try to think about that—like a kickdrum and a bop bap boop synth evolving into a washed-out reverby sound into some dry percussion. But because a lot of work on those sounds is done in real
time, I try to bring them in first before adding timbres and whatnot. I’m also trying to listen and follow on screen, maybe plan, but sometimes that breaks the flow a bit. (Interviewee D, September 2017)

The ontological expertise detailed by the account above resonates with accounts of repair work discussed earlier in the article. Kylie Jarrett (2017, p. 35) argues, for example, that the ‘agency, power and status’ of female industrial machinists were historically tied to their ‘embodiment of skill and knowledge’, and it was this connection between embodiment and expertise that ultimately led to their ‘deskilling’ during industrial revolutions and the wider turn to quantification. In the account above, expertise is located at the site of the ontological—not as authorship, agency or control, not as input that transparently produces (measurable) output. This is not a linear, agential process, but multiple temporalities and interdependencies that the live coder is living within and through. Indeed, as the live coder goes on to say, ‘in the end, I just let the improvisation take me somewhere’.

**don’t touch my MIDI cables**

All of the accounts of live coding detailed in this article problematise long-standing and normative masculinised conceptions of technology and the body, which, within the scholarship on live coding (and as detailed here), have manifested as a particular orientation towards creative and authorial agency over and through technology. These conceptions are problematic because they repeat masculine (and as scholars such as Benjamin [2019], Noble [2017] and Virginia Eubanks [2018] remind us, white and Western) ideologies through (amongst other things) valorisations of speed and code, agency and authorship. This in turn understands technology in oversimplified ways as a tool or facilitator, or problematically as a benign and apolitical support for the agential human. Such conceptions bear little resemblance to accounts of and experiences of live coding discussed in this article, which are much more attuned to, we argue, the notion of ‘creative process[es] of fashioning care and reciprocity’ that Benjamin (2018, p. 65) discusses in her definition of kinship. Our research reminds us of the continual need to push back against normative and problematic conceptions of technology and bodies, but it also—through the empirical work—highlights what is at stake if we do not. The thoughtful, creative, embodied techno–human relations, imaginings and discourse articulated by the women as kinship and as entanglements—that are also and always speaking to positionalities, identities and embodiment—make us ask questions about power and ideologies beyond the specific context of live coding.

‘Don’t touch my MIDI cables’ is not only a popular refrain from the female live coding community, who experience ‘mansplaining’ and demands to verify technological expertise. It is also a demand from scholarship, not least because cables, technologies, code and systems are not separate, fetishised or ‘owned’. They are entwined with and embedded within experiences and understandings of live coding—something more akin to ‘becoming with’ or ‘making kin’ (Kafer, 2019, p. 6). This in turn shifts focus from the potentialities of technology on or through the body to the generative capacities of mediation (for want of a better word), which are lived. Technology-as-kinship reconfigures power relations not least because we have to think of affective connections, extensions, care and ethics. It also intervenes into our imaginaries of technology, disrupting and problematising understandings of technology as a tool through which whole able-bodied white male subjectivity plays out. Kin insists not only on ‘networked interdependencies’ (ibid., p. 6) and on ‘deep affective ties’ (Rodríguez, 2014, p. 49) that are always already beyond normative understandings of both technology and subjectivity. Kinship is a ‘way of being
orientated to relations’ and through that it ‘affirms, disrupts, dismantles, regenerates and resists’ (Murphy, 2017, p. 124). These are precisely the experiences recounted here, we think, where live coding is understood in relation to multiple temporalities, sonic and technological kinship and code ideologies (for example). These are the experiences seeking to be recognised by the women live coders represented here. In this scenario, touching someone else’s MIDI cables takes on a very different meaning not least if the cables are kin with the (body of) the live coder. Far from a normatively acceptable intervention, touching someone else’s cables is political, violent and intrusive.

author biographies

Joanne Armitage is an internationally known live coder and digital artist as well as a postdoctoral researcher at the University of Cambridge, UK. She was awarded the Daphne Oram Award for Digital Innovation (2016) and the Francis Chagrin Award (2014). She has worked extensively as a creative sound and media practitioner, developing bespoke installations, commissions and performances with artists across the globe. Her current research and publications explore how gender emerges in the open-source coding scene Algorave. eenjla@leeds.ac.uk

Helen Thornham is an Associate Professor in Digital Cultures at the University of Leeds, UK. Her research focuses on gender and technological mediations, data and digital inequalities, embodiment, youth, space, place and communities. She has led a number of projects investigating practices in digital media that are funded across Research Councils UK (RCUK). She is author of Ethnographies of the Videogame: Narrative, Gender and Praxis (London and New York: Routledge, 2011) and Gender and Digital Culture: Between Irreconcilability and the Datalogical (London and New York: Routledge, 2018) as well as co-editor of Renewing Feminisms: Radical Narratives, Fantasies and Futures in Media (London and New York: I.B. Tauris, 2013) and Content Cultures (London and New York: I.B. Tauris, 2014). h.thornham@leeds.ac.uk

references

Armitage, J., 2018. Spaces to fail in: negotiating gender, community and technology in Algorave. Dancecult, 10(1), pp. 31–45.

Amrani, I. and Payne-Frank, N., 2017. Run the code: is Algorave the future of dance music? The Guardian, 30 November. Available at: https://www.theguardian.com/music/video/2017/nov/30/run-the-code-is-algorave-the-future-of-dance-music [last accessed 24 October 2019].

Andrejevic, M., 2011. The work that affective economics does. Cultural Studies, 2(4–5), pp. 604–620.

Bassett, C., Kember, S. and O’Riordan, K., 2019. Furious: Technological Feminism and Digital Futures. London: Pluto Press.

Benjamin, R., 2018. Black afterlives matter: cultivating kinfulness as reproductive justice. In A. Clarke and D. Haraway, eds. Making Kin Not Population. London: Prickly Paradigm Press, pp. 41–67.

Benjamin, R., 2019. Race After Technology: Abolitionist Tools for the New Jim Code. Cambridge: Polity Press.

Berry, D., 2014. Post-digital humanities: computation and cultural critique in arts and humanities. Educause, 49(3), pp. 22–26.

Bolt, I., 2017. Meet the female coders pushing electronic music into the future. Mixmag, 31 August. Available at: https://mixmag.net/feature/female-coders-algorave [last accessed 10 August 2020].

Braidotti, R., 2013. The Posthuman. Cambridge: Polity Press.
Burch, S., 2014. Dislocated histories: the canton asylum for insane Indians. *Women, Gender, and Families of Color, 2*(2), pp. 141–162.

Cardenas, A., 2018. Live coding: a new approach to musical composition. *Dancecult, 10*(1). Available at: http://dx.doi.org/10.12801/1947-5403.2018.10.01.14 [last accessed 10 August 2020].

Carr, D., 2002. Playing with Lara. In G. King and T. Krzywinska, eds. *Screenplay: Cinema/Videogames/Interfaces*. London: Wallflower Press, pp. 171–180.

Castells, M., 2010 [1996]. *The Rise of the Network Society*. Malden: Blackwell.

Clarke, A., 2018. Introducing making kin not population. In A. Clarke and D. Haraway, eds. *Making Kin Not Population*. London: Prickly Paradigm Press, pp. 1–41.

Clarke, A. and Haraway, D., eds., 2018. *Making Kin Not Population*. London: Prickly Paradigm Press.

Cockburn, C., 1992. The circuit of technology gender, identity and power. In E. Hirsch and R.H. Silverstone, eds. *Media and Information in Domestic Spaces*. London: Routledge, pp. 32–48.

Cocker, E., 2016. Performing thinking in action: the meletē of live coding. *International Journal of Performance Arts and Digital Media, 12*(2), pp. 102–116.

Cocker, E., 2018. What now, what next—kairotic coding and the unfolding future seized. *Digital Creativity, 29*(1), pp. 82–95.

Collins, N., McLean, A., Rohrhuber, J. and Ward, A., 2003. Live coding in laptop performance. *Organised Sound, 8*(3), pp. 321–350.

Coulson, N. and Mejias, U.A., 2019. Data colonialism: rethinking big data’s relation to the contemporary subject. *Television and New Media, 20*(4), pp. 336–349.

Cox, G. and McLean, A., 2013. *Speaking Code: Coding as Aesthetic and Political Expression*. Cambridge: MIT Press.

Day, S. and Lury, C., 2016. Biosensing: tracking persons. In D. Nafus, ed. *Quantified: Biosensing Technologies in Everyday Life*. Cambridge: MIT Press, pp. 43–67.

Eubanks, V., 2018. *Automating Inequality: How High-Tech Tools Profile, Punish and Police the Poor*. New York: St Martins Press.

*Generation Algorave*, 2019. Film. Directed by J. Sansom. UK: Resident Advisor. Available at: https://www.residentadvisor.net/features/3396 [last accessed 20 February 2020].

Gómez Cruz, E. and Thornham, H., 2015. ‘Raw talent in the making’: imaginary journeys, authorship and the discourses of expertise. *Convergence, 21*(3), pp. 314–327.

Grosz, E., 2001. *Architecture from the Outside: Essays on Virtual and Real Spaces*. Cambridge: MIT Press.

Grosz, E., 2005. *Time Travels: Feminism, Nature, Power*. Durham: Duke University Press.

Halberstam, J., 2005. *In a Queer Time and Place: Transgender Bodies, Subcultural Lives*. New York: New York University Press.

Hansen, M.B.N., 2009. Living (with) technical time: from media surrogacy to distributed cognition. *Theory, Culture and Society, 26*(2–3), pp. 294–315.

Haraway, D., 2008. *When Species Meet*. Minneapolis: University of Minnesota Press.

Haraway, D., 2016. *Staying with the Trouble: Making Kin in the Chthulucene*. Durham: Duke University Press.

Hayles, N.K., 2017. *Unthought: The Power of the Cognitive Nonconscious*. Chicago: Chicago University Press.

Hicks, M., 2018. *Programmed Inequalities: How Britain Discarded Women Technologists and Lost Its Edge in Computing*. Cambridge: MIT Press.
Jackson, S.J., 2017. Speed, time, infrastructure: temporalities of breakdown, maintenance, and repair. In J. Wajcman and N. Dodd, eds. The Sociology of Speed: Digital, Organizational and Social Temporalities. Oxford: Oxford University Press, pp. 169–187.

Jarrett, K., 2017. Feminism, Labour and Digital Media: The Digital Housewife. London: Routledge.

Kafer, A., 2019. Crip, kin, manifesting. Catalyst: Feminism, Theory, Technoscience, 5(1), pp. 1–37.

Lewis, G.E., 2018. Why do we want our computers to improvise? In A. McLean and R.T. Dean, eds. The Oxford Handbook of Algorithmic Music. Oxford: Oxford University Press, pp. 123–132.

Magnusson, T., 2011. Algorithms as scores: coding live music. Leonardo Music Journal, 21, pp. 19–23.

Mattern, S.C., 2017. Code + Clay ... Data + Dirt: Five Thousand Years of Urban Media. Minneapolis: University of Minnesota Press.

McLean, A. and Dean, R.T., eds., 2018. The Oxford Handbook of Algorithmic Music. New York: Oxford University Press.

McNeil, M., ed., 1987. Gender and Expertise. London: Free Association Books.

McNay, L., 2000. Gender and Agency: Reconfiguring the Subject in Feminist and Social Theory. Cambridge: Polity Press.

Murphy, M., 2017. Against population, towards alterlife. In A. Clarke and D. Haraway, eds. Making Kin Not Population. London: Prickly Paradigm Press, pp. 101–125.

Noble, S., 2017. Algorithms of Oppression. New York: New York University Press.

Parkinson, A. and Bell, R., 2015. Deadmau5, Derek Bailey, and the laptop instrument—improvisation, composition, and liveness in live coding. In A. McLean, T. Magnusson, K. Ng, S. Knotts and J. Armitage, eds. Proceedings of the First International Conference on Live Coding, ICSRiM, University of Leeds, Leeds, UK. Leeds: ICSRiM, School of Music, University of Leeds. pp. 170–178. Available at: https://zenodo.org/record/19350#.XlTlcS10fOQ [accessed 20 February 2020].

Ricourte, P., 2019. Data epistemologies: the coloniality of power, and resistance. Television and New Media, 20(4), pp. 350–365.

Roberts, C. and Kuchera-Marin, J., 2012. Gibber: live coding audio in the browser. Conference paper presented at International Conference on Computer Music (ICMC) 2012: Non-cochlear sound, 9–14 September. Ljubljana. Available at: https://quod.lib.umich.edu/cgi/p/psdb/id/zoom/iom/cdd2372.2012.011;form=pdf [last accessed 24 February 2020].

Rodgers, T., 2010. Pink Noises: Women on Electronic Music and Sound. Durham: Duke University Press.

Rodríguez, J.M., 2014. Sexual Futures, Queer Gestures, and Other Latina Longings. New York: New York University Press.

Suchman, L., 2007. Agencies in technology design: feminist reconfigurations. Working paper, CiteSeerX. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.548.7968 [accessed 2 December 2020].

Thornham, H., 2019. Gender and Digital Culture: Between Irreconcilability and the Datalogical. London: Routledge.

Thornham, S., 2007. Women, Feminism and Media. Edinburgh: Edinburgh University Press.

Urry, J., 2000. Sociology Beyond Societies: Mobilities for the Twenty-First Century. New York: Routledge.

Wajcman, J., 2004. Technofeminism. Cambridge: Polity Press.

Wajcman, J., 2019. The digital architecture of time management. Science, Technology & Human Values, 44(2), pp. 315–337.

Wajcman, J. and Dodd, N., eds., 2017. The Sociology of Speed: Digital, Organizational and Social Temporalities. Oxford: Oxford University Press.

White, E.C., 1987. Kaironomia: On the Will to Invent. Ithaca: Cornell University Press.